



MASTERARBEIT

Frau
Katharina Schönefeld

Unterstützung der Vor-Ort- Dokumentation zur qualitativen Verbesserung von Tätigkeitsnachweisen in der dualen Berufsausbildung

Konzeption und Realisierung einer auch für mobile
Endgeräte geeigneten Layout-Struktur auf Basis des
Online-Berichtsheftes BLok unter Beachtung damit
verbundener Aspekte der Usability

2012

MASTERARBEIT

Unterstützung der Vor-Ort-Dokumentation zur qualitativen Verbesserung von Tätigkeits- nachweisen in der dualen Berufsausbildung

Konzeption und Realisierung einer auch für mobile Endgeräte geeigneten
Layout-Struktur auf Basis des Online-Berichtsheftes BLok unter Beachtung
damit verbundener Aspekte der Usability

Autorin:

Katharina Schönefeld

Studiengang:

Industrial Management

Seminargruppe:

ZM10wT-M

Erstprüfer:

Prof. Dr.-Ing. Robert J. Wierzbicki

Zweitprüfer:

Dipl. Wirt.-Inf. Lars Eberle

MASTER THESIS

Support of on-site documentation for quality improvement of activity reports in vocational trainings

Conception and implementation of a layout structure suitable for
mobile devices, based on the online activity report tool BLok and
by taking into account related usability issues

Author:

Katharina Schönefeld

Course of studies:

Industrial Management

Seminar group:

ZM10wT-M

First examiner:

Prof. Dr.-Ing. Robert J. Wierzbicki

Second examiner:

Dipl. Wirt.-Inf. Lars Eberle

Bibliografische Angaben

Schönefeld, Katharina:

Unterstützung der Vor-Ort-Dokumentation zur qualitativen Verbesserung von Tätigkeitsnachweisen in der dualen Berufsausbildung. Konzeption und Realisierung einer auch für mobile Endgeräte geeigneten Layout-Struktur auf Basis des Online-Berichtsheftes BLok unter Beachtung damit verbundener Aspekte der Usability. - 2012. - 177 Seiten, 96 Abbildungen, 14 Tabellen, 4 Anlagen

Mittweida, Hochschule Mittweida (FH), Institut für Technologie- und Wissenstransfer Mittweida
Masterarbeit, 2012

Referat

Das Ziel der vorliegenden Masterarbeit ist die Neu-Konzeption und Umsetzung der Anwendung BLok, die sowohl benutzerfreundlich und zugänglich, als auch auf allen Geräten verwendbar sein soll. Bei BLok handelt es sich um ein online bereitgestelltes Berichtsheft, welches durch Auszubildende gepflegt und durch die zugehörigen Ausbilder kontrolliert werden kann. Die Verwendbarkeit auf mobilen Geräten soll die Dokumentation des Ausbildungsnachweises direkt am Ausbildungsort weiter verbessern und fördern.

Als Basis für die Neu-Konzeption dienen Recherchen zu den Themen Usability und Accessibility für Webanwendungen mit Fokus auf der mobilen Nutzung. Dabei wird zwischen einer Webanwendung für mobile Geräte, die über den Browser aufgerufen wird und einer nativen App für jedes Betriebssystem unterschieden. Aufgrund der Optimierung für mobile Geräte ergeben sich weiterhin technische Besonderheiten und neue Konzepte zur Umsetzung, die in die Untersuchung ebenfalls mit einfließen.

Um neben den theoretischen Grundlagen und Möglichkeiten zur technischen Umsetzung außerdem einen Bezug zur Praxis herzustellen, werden ausgewählte Anwendungen analysiert. Diese Untersuchung wird anhand der Richtlinien für Usability und Accessibility durchgeführt. Bei den Anwendungen handelt es sich um das Online-Notizbuch Evernote und die E-Mail Anwendung Google Mail.

Mit Hilfe der gewonnenen Erkenntnisse aus der Theorie und der Analyse wird die Benutzeroberfläche der Anwendung BLok konzipiert. Dabei wird zwischen einer Variante der Webanwendung für den Desktop und für mobile Geräte sowie einer nativen App unterschieden. Auf Basis des Konzeptes entstehen ein Styleguide für BLok sowie grafische Entwürfe für die verschiedenen Arten der Anwendung. Im Anschluss erfolgt die Umsetzung der Benutzeroberfläche von BLok mit Hilfe der Technologien HTML und CSS. Abschließend wird der entstandene Prototyp der Anwendung anhand der Kriterien Usability und Accessibility untersucht. Auf diese Weise können die Umsetzung bewertet und weitere Verbesserungen identifiziert werden.

Inhaltsverzeichnis

Inhaltsverzeichnis	IX
Abbildungsverzeichnis	XI
Tabellenverzeichnis.....	XV
Abkürzungsverzeichnis	XVII
1 Einleitung.....	19
1.1 Problemstellung	19
1.2 Zielsetzung.....	20
1.3 Vorgehensweise.....	21
2 Theoretische Grundlagen zu Usability und Accessibility	23
2.1 Usability - Richtlinien für die Benutzerfreundlichkeit	23
2.1.1 Usability allgemein	23
2.1.2 Mobile Usability.....	31
2.1.3 Usability nativer Apps.....	40
2.2 Accessibility - Richtlinien für die Zugänglichkeit.....	52
2.2.1 Überblick	52
2.2.2 Web Content Accessibility Guidelines (WCAG) 2.0	52
2.2.3 Zugänglichkeit von Geräten mit Touchscreen	55
3 Moderne Webtechnologien	57
3.1 HTML5	57
3.1.1 Die HTML5-Syntax.....	57
3.1.2 Sectioning Elemente	58
3.1.3 Offline Web Applications	59
3.1.4 Geolocation API	61
3.2 Mobile Webseite oder native App.....	62
3.3 Technologien für das mobile Web	64
3.3.1 Separate mobile Webseite	64
3.3.2 Eine Webseite für alle Geräte	65
3.3.3 Web-Apps	73
3.4 Technologien zur Erstellung von nativen Apps.....	76
3.4.1 Entwicklung herstellerspezifischer Apps	76
3.4.2 Herstellerunabhängige Erstellung mit PhoneGap	77
3.5 Technologien für die Zugänglichkeit.....	78
3.5.1 WAI-ARIA.....	78
3.5.2 Zugänglichkeit von nativen Apps.....	81

4 Analyse und Bewertung ausgewählter Anwendungen	89
4.1 Auswahl der Anwendungen.....	89
4.2 Evernote - Online Notizbuch	90
4.2.1 Bewertung nach ISO 9241-110 Dialoggestaltung	91
4.2.2 Bewertung Usability für mobile Geräte.....	97
4.2.3 Zugänglichkeit der Evernote App	100
4.2.4 Zusammenfassung Evernote	102
4.3 Google Mail	103
4.3.1 Bewertung nach ISO 9241-110 Dialoggestaltung	103
4.3.2 Bewertung Usability für mobile Geräte.....	108
4.3.3 Zugänglichkeit der Google Mail App	112
4.3.4 Zusammenfassung Google Mail	114
4.4 Zusammenfassung der Analyse.....	114
5 Konzeption Benutzeroberfläche und Bedienung von BLok	115
5.1 Allgemeine Ansprüche an das Konzept	115
5.2 Konzept für die Benutzeroberfläche.....	116
5.2.1 Konzept für die mobile Anwendung.....	116
5.2.2 Konzept für die iOS App.....	120
5.3 Das Bedienkonzept für BLok.....	121
5.3.1 Bedienung der mobilen Webanwendung	122
5.3.2 Bedienung der iOS App.....	124
6 Gestalterische und technische Umsetzung der Anwendung BLok	127
6.1 Gestalterische Umsetzung	127
6.1.1 Entwicklung eines Styleguides für BLok	127
6.1.2 Gestaltung der Webanwendung.....	129
6.1.3 Gestaltung der nativen App.....	132
6.2 Technische Umsetzung	133
6.2.1 Webanwendung BLok	133
6.2.2 Anpassungen für die mobile Webseite.....	136
6.3 Evaluation der neuen Anwendung BLok	140
6.3.1 Bewertung nach ISO 9241-110 Dialoggestaltung	140
6.3.2 Bewertung Usability für mobile Geräte.....	143
6.3.3 Zugänglichkeit der Anwendung BLok.....	145
6.3.4 Zusammenfassung BLok	146
7 Zusammenfassung und Ausblick.....	147
Anlagen	151
Literaturverzeichnis	167
Erklärung zur selbstständigen Anfertigung.....	177

Abbildungsverzeichnis

Abbildung 1	Übersicht über ISO 9241-11	24
Abbildung 2	Hauptnavigation am unteren Bildschirmrand	34
Abbildung 3	Iceberg Tip	37
Abbildung 4	Call-To-Action unter iOS	38
Abbildung 5	Mail Anwendung unter iOS, Android und Windows Phone 7	41
Abbildung 6	Windows Phone 7: Orientierungen	43
Abbildung 7	Startbildschirm von iOS, Android und Windows Phone 7 im Vergleich	44
Abbildung 8	Navigation Bar unter iOS, Android und Windows Phone 7	44
Abbildung 9	Toolbar unter iOS, Android und Windows Phone 7	45
Abbildung 10	Tab Bar unter iOS und Android	46
Abbildung 11	Pivot Control unter Windows Phone 7	47
Abbildung 12	Panorama Control unter Windows Phone 7	48
Abbildung 13	Listen und Tabellen unter iOS, Android und Windows Phone 7	48
Abbildung 14	Grid List unter Android	49
Abbildung 15	Actions Sheets unter iOS und Auswahlliste unter Android	50
Abbildung 16	Action Sheet auf dem iPhone	50
Abbildung 17	Formulare und Texteingaben unter iOS und Windows Phone 7	51
Abbildung 18	Formulare und Texteingaben unter Android	51
Abbildung 19	Auszeichnung der Sprache über das Attribut lang	55
Abbildung 20	Doctype-Definition: HTML 4.01 und HTML5 im Vergleich	57
Abbildung 21	HTML5: Reduziertes Dokument.....	58
Abbildung 22	HTML5: CSS-Anweisung für die Unterstützung der Sectioning Elemente	59
Abbildung 23	HTML5: Javascript-Anweisung für den Internet Explorer	59
Abbildung 24	HTML5: Offline Web Applications - Manifest	60
Abbildung 25	HTML5: Offline Web Applications - HTML-Datei.....	60
Abbildung 26	HTML5: Offline Web Applications - Fallback und Network	61
Abbildung 27	Beispiel Responsive Webdesign.....	67
Abbildung 28	Media Queries.....	68
Abbildung 29	Media Queries für Standard Geräte.....	69
Abbildung 30	Nutzung von Adapt.js	71
Abbildung 31	Webdesign mit RESS	72
Abbildung 32	Dynamischer Header und Footer.....	73
Abbildung 33	Angepasste Auslieferung von Bildern	73
Abbildung 34	Web-App Google Mail.....	74
Abbildung 35	Einbindung von jQuery Mobile	75
Abbildung 36	Metadatum Viewport.....	75

Abbildung 37	Auszeichnung einer HTML-Datei mit data-role Attributen	75
Abbildung 38	jQuery Mobile im Einsatz	76
Abbildung 39	WAI-ARIA: Verwendung der Document Landmark Roles	79
Abbildung 40	WAI-ARIA: Verwendung von aria-labelledby und aria-describedby	80
Abbildung 41	Einsatz von VoiceOver unter iOS	82
Abbildung 42	Accessibility im Interface Builder	84
Abbildung 43	Definition von Attributen für die Zugänglichkeit unter iOS	84
Abbildung 44	Rotor unter iOS	85
Abbildung 45	Hinzufügen eines Labels unter Android	86
Abbildung 46	Setzen der Fokus-Reihenfolge unter Android	87
Abbildung 47	BLok: Ansicht Berichtsheft	89
Abbildung 48	Evernote: Ausschnitt der Webanwendung	91
Abbildung 49	Evernote: Übersicht Notizbücher	92
Abbildung 50	Evernote: Übersicht über Notizen auf dem iPad	93
Abbildung 51	Evernote: Neue Notiz anlegen	93
Abbildung 52	Evernote: Rückmeldung über Synchronisation einer Notiz	94
Abbildung 53	Evernote: Hinweise auf dem iPhone und iPad	95
Abbildung 54	Evernote als Webanwendung und als native App unter iOS	97
Abbildung 55	Evernote: Navigation zwischen den Ansichten auf dem iPhone und iPad ..	98
Abbildung 56	Evernote: Platzierung der Buttons	99
Abbildung 57	Evernote: Eingabefelder	100
Abbildung 58	Evernote: Verwendung von VoiceOver unter iOS	101
Abbildung 59	Evernote: Detailansicht einer Notiz	102
Abbildung 60	Google Mail: Optionen Posteingang und Detailansicht E-Mail	104
Abbildung 61	Google Mail: Menü auf dem iPhone und iPad	104
Abbildung 62	Google Mail: Rückmeldungen und Hinweise	105
Abbildung 63	Google Mail: Hinweismeldung bei fehlerhafter Eingabe	107
Abbildung 64	Google Mail: Native App für iOS und mobile Web-App im Vergleich	108
Abbildung 65	Mail-App von Apple und Google im Vergleich	109
Abbildung 66	Google Mail: Navigation zwischen einzelnen Postfächern	109
Abbildung 67	Google Mail: Gestaltung der Interaktionselemente	111
Abbildung 68	Google Mail: E-Mail Formular	111
Abbildung 69	Google Mail: Verwendung von VoiceOver unter iOS - Ansicht 1	112
Abbildung 70	Google Mail: Verwendung von VoiceOver unter iOS - Ansicht 2	113
Abbildung 71	BLok: Entwurf für das Berichtsheft	117
Abbildung 72	BLok: Ansicht Berichtsheft	117
Abbildung 73	BLok: Entwurf mobile Webanwendung - Smartphone	118
Abbildung 74	BLok: Entwurf mobile Webanwendung - Tablet	120
Abbildung 75	BLok: Entwurf iOS App	121

Abbildung 76	BLok: Verwendete Gesten	122
Abbildung 77	BLok: Bedienung der Webanwendung	123
Abbildung 78	BLok: Bedienung der iOS App - Ansicht 1	124
Abbildung 79	BLok: Bedienung der iOS App - Ansicht 2	125
Abbildung 80	Logo BLok - Das Online-Berichtsheft.....	127
Abbildung 81	BLok: Primäre und sekundäre Farben	128
Abbildung 82	BLok: Icons für die Navigation und für ausgewählte Funktionen	129
Abbildung 83	BLok: Layout Kopfbereich und Navigation.....	129
Abbildung 84	BLok: Entwurf Kopfbereich der mobilen Version	130
Abbildung 85	BLok: Entwurf Navigation der mobilen Version.....	130
Abbildung 86	BLok: Gestaltung der Startseite	130
Abbildung 87	BLok: Gestaltung des Berichtshefts.....	131
Abbildung 88	BLok: Berichtsheft in der mobilen Version	132
Abbildung 89	BLok: Gestaltung der App für iOS.....	133
Abbildung 90	BLok: Ausschnitt der HTML-Struktur.....	134
Abbildung 91	BLok: Angepasstes Layout für Thyssen Krupp	135
Abbildung 92	BLok: Verwendete Media Queries	137
Abbildung 93	BLok: Header bei 1250 Pixeln und 999 Pixeln Breite	137
Abbildung 94	BLok: Layout bei 655 Pixeln und bei 450 Pixeln Breite	138
Abbildung 95	BLok: Auswahl der Zeit im Berichtsheft	144
Abbildung 96	BLok: Verwendung von VoiceOver unter iOS	145

Tabellenverzeichnis

Tabelle 1	Eigenschaften von Informationen	30
Tabelle 2	Organisation von Informationen.....	30
Tabelle 3	Kodierung von Informationen.....	31
Tabelle 4	Gesten für die Bedienung eines Touchscreens	36
Tabelle 5	WCAG 2.0: Prinzip 1 - Wahrnehmbar	53
Tabelle 6	WCAG 2.0: Prinzip 2 - Bedienbar	54
Tabelle 7	WCAG 2.0: Prinzip 3 - Verständlich	54
Tabelle 8	WCAG 2.0: Prinzip 4 - Robust	55
Tabelle 9	HTML5: Übersicht Sectioning Elemente	58
Tabelle 10	Eigenschaften für Media Queries.....	68
Tabelle 11	WAI-ARIA: Document Landmark Roles	79
Tabelle 12	WAI-ARIA: Ausgewählte Zustände und Eigenschaften	79
Tabelle 13	WAI-ARIA: Live Regions.....	81
Tabelle 14	Attribute für die Auszeichnung von Elementen unter iOS	83

Abkürzungsverzeichnis

API	Application Programming Interface
ARIA	Accessible Rich Internet Applications
BGG	Behindertengleichstellungsgesetz
BLok	Online-Berichtsheft zur Stärkung der Lernortkooperation
CSS	Cascading Style Sheets
DIN	Deutsches Institut für Normung
DOM	Document Object Model
EN	European Norm
GPS	Global Positioning System
HTML	Hypertext Markup Language
ISO	International Organization for Standardization
RESS	Responsive Web Design with Server Side Components
W3C	World Wide Web Consortium
WAI	Web Accessibility Initiative
WCAG	Web Content Accessibility Guidelines
WHATWG	Web Hypertext Application Technology Working Group

1 Einleitung

1.1 Problemstellung

Durch den technischen Fortschritt sind heute in nahezu allen Lebensbereichen Computer mit Internetverbindung zu finden. Wir nutzen diese und das Internet nicht nur auf Arbeit, sondern auch in unserer Freizeit, immer öfter auch unterwegs und somit ortsunabhängig. Computer können uns dabei effektiv unterstützen, sodass wir unter anderem Zeit und Geld sparen.

In Unternehmen, Büros und Schulen entstehen heute große Mengen von Dokumenten, die bearbeitet, aktualisiert, sortiert und archiviert werden müssen. Der damit verbundene erhöhte Verwaltungsaufwand kostet Zeit und Geld. Computer ermöglichen in diesem Zusammenhang eine nahezu papierlose Erfassung und Verwaltung von Daten. Auf diese Weise lässt sich Zeit bei der Erfassung, Aktualisierung und beim Durchsuchen der Dokumente sparen. Weiterhin ist eine effektive und effiziente Zusammenarbeit durch den orts- und zeitunabhängigen Austausch der Dokumente über das Internet möglich. Mit Hilfe von Computern kann außerdem für eine optimierte Weiterverarbeitung der Dokumente gesorgt werden, da diese bereits digital vorliegen. Auch Probleme mit der Lesbarkeit aufgrund einer individuellen Handschrift werden durch das digitale Erfassen von Informationen vermieden. Durch organisiertes Ablegen und bestimmte Sicherheitsmaßnahmen kann das Verlorengehen von wichtigen Dokumenten verhindert werden.

Um speziell in Berufsschulen eine papierlose Erfassung von Informationen zu ermöglichen, wurde im Rahmen eines Hochschulprojektes an der Technischen Universität Dresden die Anwendung „BLok - Das Online-Berichtsheft“ entwickelt, welches durchgängig auf Online-Technologien setzt. Die Auszubildenden führen ihr Berichtsheft im Internet und können dieses Ausbilden und Berufsschullehrern ebenfalls online zur Abnahme vorlegen. Selbst die Abschlussprüfung kann komplett papierlos erfolgen. Weiterhin stehen für die Kommunikation zwischen Ausbilden und Auszubildenden verschiedene Funktionen zur Verfügung. Auf diese Weise können sich die Auszubildenden Rückmeldungen zu ihrem Berichtsheft holen oder von den Ausbilden eingeschätzt werden.

Die Erfassung der Tätigkeitsnachweise über die Webanwendung BLok ermöglicht neben der Zeitersparnis auch eine verbesserte Kommunikation und Zusammenarbeit von Auszubildenden, Ausbilden und Berufsschullehrern. Fortschritte der Auszubildenden, neue Dokumente und die Anmerkungen der Ausbilder sind sofort zugänglich. Durch das Hinzufügen von Bemerkungen, Notizen und ausbildungsrelevanten Dokumenten wird eine Diskussion über den aktuellen Bericht unterstützt. Weiterhin stehen in BLok Kommunikationsmittel, wie zum Beispiel Nachrichten zum Austausch zwischen den Beteiligten zur Verfügung.

Ein besonderer Vorteil von BLok ist die Möglichkeit, die Tätigkeitsnachweise zu jeder Zeit und an jedem Ort zu bearbeiten. Auszubildende und Ausbilder sind somit flexibler und wichtige Informationen können zeitnah oder sogar direkt vor Ort erfasst werden.

Um die Erfassung von Tätigkeitsnachweisen mit Hilfe von Online-Technologien zu fördern, sollte die Anwendung BLok zur Nutzung motivieren. Eine einfache und intuitive Bedienung muss aus diesem Grund gegeben sein. Die Benutzer sollten ihre jeweiligen Aufgaben effizient und effektiv erledigen können und die Nutzung der Anwendung muss zufriedenstellend sein. Sind diese Ansprüche erfüllt, beurteilt der Benutzer die Anwendung positiv und wird zum erneuten Nutzen motiviert.

Diese Ansprüche lassen sich unter dem Begriff Benutzerfreundlichkeit (engl. Usability) zusammenfassen. Weiterhin muss die Anwendung auch für Menschen mit Behinderungen benutzbar und somit barrierefrei sein. Zu berücksichtigende Behinderungen sind dabei nicht nur Schwierigkeiten mit dem Sehsinn, sondern auch Einschränkungen die Motorik, das Gehör und die Kognition betreffend.

Zusätzlich sollte BLok auf allen Geräten benutzbar sein, zum Beispiel auf mobilen Endgeräten mit Touchscreen. An dieser Stelle sind nicht nur der abweichende Funktionsumfang der Geräte zu beachten, sondern auch veränderte Bedienparadigmen und damit verbundene Besonderheiten bei Usability und Accessibility.

1.2 Zielsetzung

Im Rahmen der vorliegenden Arbeit wird die Benutzerfreundlichkeit und Zugänglichkeit von Webanwendungen, die auf allen Geräten funktionieren, untersucht. Es ist zu klären, wie eine Webanwendung gestaltet und strukturell aufgebaut sein muss, um diese Anforderungen zu erfüllen. Die gewonnenen Erkenntnisse werden auf BLok übertragen und somit praktisch angewendet.

Das Hauptziel der Masterarbeit ist dabei die Untersuchung der Benutzerfreundlichkeit von Webanwendungen für mobile Endgeräte mit Touchscreen. Primär wird bei deren Nutzung auf mobilen Endgeräten zwischen mobilen Anwendungen und nativen Apps unterschieden. Eine mobile Webseite wird über einen Browser auf dem Endgerät aufgerufen. Eine native App muss hingegen speziell für das jeweilige Betriebssystem programmiert und auf dem Endgerät vom Benutzer installiert werden. Die jeweiligen Besonderheiten bei der Gestaltung und Bedienung werden herausgearbeitet.

Um die Zugänglichkeit einer Webanwendung sicherzustellen sind zusätzliche Richtlinien, wie zum Beispiel die Web Content Accessibility Guidelines (WCAG)¹ zu beachten und umzusetzen. Bei den Richtlinien der WCAG handelt es sich um eine Spezifikation des World Wide Web Consortiums (W3C), welches grundlegende Schritte und Technologien definiert, um Webseiten und Webanwendungen für alle Benutzer zugänglich zu gestalten. Bei der Nutzung von mobilen Endgeräten gibt es weitere Besonderheiten, die im Rahmen der Arbeit ebenfalls untersucht werden.

Aufgrund der Vielfalt von Endgeräten und des schnellen technologischen Fortschritts müssen Webseiten und Anwendungen möglichst flexibel und leicht anzupassen sein. Es ist zu beachten, dass eine Anwendung sowohl für die Nutzung mit einem Rechner, als auch mit Touchscreens unterschiedlicher Größe zu optimieren ist. Um diese Forderung zu erfüllen werden moderne Webtechnologien und Konzepte untersucht und evaluiert.

Die Ergebnisse der Untersuchungen zu Usability und Accessibility werden auf die Anwendung BLok der BPS Bildungsportal Sachsen GmbH übertragen. BLok muss sowohl bei der Nutzung mit einem Rechner als auch mit mobilen Endgeräten benutzerfreundlich und intuitiv sein. Daher werden Benutzeroberfläche und Struktur der Anwendung optimiert und an diese Anforderungen angepasst. Im Rahmen der Zugänglichkeit sind für BLok vor allem Nutzer mit kognitiven Behinderungen zu berücksichtigen.

1 W3C: Web Content Accessibility Guidelines (WCAG) 2.0

Das Ergebnis der Masterarbeit ist die Vorbereitung der Umstellung von BLok auf das neue benutzerfreundliche und zugängliche Layout, sodass die Anwendung im Browser und auf mobilen Endgeräten nutzbar ist. Zu diesem Zweck entsteht ein Prototyp, der die Benutzeroberfläche abbildet und somit die zukünftige Anwendung veranschaulicht. Beispielhaft wird zudem ein alternatives Layout für einen Kunden umgesetzt, um die leichte Anpassbarkeit zu demonstrieren.

1.3 Vorgehensweise

Zunächst werden Grundlagen zu den Themen Usability und Accessibility erarbeitet. Neben einer allgemeinen Übersicht liegt der Schwerpunkt auf den Besonderheiten bei der Bedienung von mobilen Endgeräten mit Touchscreen. Bei nativen Apps sind seitens der Hersteller der unterschiedlichen Betriebssysteme Richtlinien und Gestaltungshinweise vorgegeben. Die Vorgaben der drei großen Hersteller Apple, Google und Microsoft werden auf Gemeinsamkeiten und Unterschiede hin untersucht und in die Grundlagen mit einbezogen.

Das Ergebnis der Recherche ist eine Zusammenstellung von Kriterien, die bei der Erstellung von nativen Apps und von Webanwendungen für die mobile Nutzung zu beachten sind.

Für die Überarbeitung einer Anwendung hinsichtlich der mobilen Benutzung sind nicht nur Optimierungen von Usability und Accessibility, sondern auch eine Modernisierung der Technologie notwendig. Daher werden in dieser Arbeit zusätzlich moderne Web-Technologien, wie zum Beispiel HTML5 und ARIA vorgestellt und deren Anwendbarkeit untersucht.

Neben den theoretischen Betrachtungen wird auch die Praxis in die Arbeit mit einbezogen. Daher werden bestehende Anwendungen ausgewählt und mit Hilfe der Kenntnisse über Usability, Accessibility und Möglichkeiten zur technischen Umsetzung untersucht und bewertet. Bei der Auswahl der Anwendungen wird insbesondere auf die Vergleichbarkeit mit BLok geachtet, um die Ergebnisse der Untersuchung sinnvoll auf BLok übertragen zu können. Die Anwendungen sollten daher über erweiterte Möglichkeiten der Inhaltseingabe und Bearbeitung verfügen. Dies trifft zum Beispiel auf das E-Mail-Programm Google Mail oder die Notizbuch-Anwendung Evernote² zu.

Ziel der Analyse der ausgewählten Anwendungen ist es, sich an den positiven Aspekten zu orientieren und diese bei Eignung für BLok zu übernehmen. Die eher negativ auffallenden Eigenschaften werden besonders beachtet und im Rahmen von BLok verbessert. Das Ergebnis der Analyse der ausgewählten Anwendungen ist ein Überblick über die gestalterischen und technischen Möglichkeiten bei der Entwicklung einer Anwendung für den mobilen Kontext.

Im Anschluss an die Analyse entsteht ein Konzept für die Anwendung BLok. Dabei wird zwischen dem Entwurf für die Webanwendung, die mobile Anwendung und die App unterschieden. Das Konzept umfasst einen Styleguide mit Gestaltungsrichtlinien für die verschiedenen Anwendungsbereiche von BLok sowie die Zusammenstellung des Bedienkonzeptes. Dies bildet somit die Grundlage für die anschließende Optimierung von BLok für die mobile Nutzung.

2 Evernote Corporation: Evernote, Ihr virtuelles Gedächtnis

Die gestalterische Umsetzung umfasst zunächst einen grafischen Entwurf der Oberfläche für die Webanwendung, für die mobile Anwendung und für die App. Für die mobile Anwendung sind besonders die unterschiedlichen Bildschirmgrößen zu beachten und in die Gestaltung des Layouts mit einzubeziehen. Beim Entwurf für die native App sind zusätzlich die spezifischen Vorgaben der Hersteller des jeweiligen Betriebssystems zu berücksichtigen.

Im Anschluss folgt auf Basis dieser Entwürfe und des Bedienkonzeptes die Implementierung der Webanwendung. Im Rahmen der Arbeit werden zu diesem Zweck die HTML-Struktur und die CSS-Dateien erstellt. Die funktionale Umsetzung wird von einem Mitarbeiter der BPS Bildungsportal Sachsen GmbH durchgeführt.

Der entstandene Prototyp der Anwendung BLok wird abschließend nach den zuvor aufgestellten Kriterien untersucht. Auf diese Weise wird sichergestellt, dass die angestrebten Verbesserungen von BLok in Bezug auf die Usability und Accessibility im mobilen Kontext auch erreicht wurden.

2 Theoretische Grundlagen zu Usability und Accessibility

2.1 Usability - Richtlinien für die Benutzerfreundlichkeit

2.1.1 Usability allgemein

Usability setzt sich aus den englischen Wörtern „to use“ (benutzen) und „the ability“ (die Fähigkeit) zusammen und wird mit Gebrauchstauglichkeit, Bedienbarkeit oder Benutzerfreundlichkeit übersetzt.³ Auch der Begriff Software-Ergonomie, speziell für interaktive Computersysteme⁴, kann in diesem Zusammenhang verwendet werden.

Die Gebrauchstauglichkeit wird wie folgt definiert:

„Das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen.“⁵

Bei der Benutzerfreundlichkeit geht es damit nicht nur um messbare Aspekte, wie die Effektivität und Effizienz bei der Nutzung eines Systems, sondern auch um eine emotionale, subjektive Erfahrung. Die vom jeweiligen Benutzer erlebte Qualität der Nutzung steht dabei im Vordergrund. Es ist wichtig, dass der Benutzer zufrieden ist und gerne mit dem System arbeitet.

Um die Benutzbarkeit von Systemen zu gewährleisten existieren Richtlinien, unter anderem zur Gestaltung und Anordnung von Bedien- und Navigationselementen, der Gliederung von Informationen, der Benutzerführung und der Orientierung.⁶

Wichtige Regeln und Kriterien sind in den Normen der Reihe DIN EN ISO 9241 Ergonomie der Mensch-System-Interaktion aufgeführt und mit Beispielen unterlegt. Im folgenden Kapitel werden relevante Normen vorgestellt.

2.1.1.1 Normen

Bei Normen handelt es sich um Dokumente, die von einer anerkannten Normungsorganisation, wie zum Beispiel dem Deutschen Institut für Normung (DIN), erstellt werden. Die International Organization for Standardization ist auf internationaler Ebene für die Normung zuständig.⁷

Die Norm ISO 9241 Ergonomie der Mensch-System-Interaktion liefert wichtige Richtlinien zur benutzerfreundlichen Gestaltung interaktiver Systeme. Sie behandelt dabei zum Beispiel die Darstellung von Informationen oder die Anforderungen an die Hardware.

3 Vgl. Handbuch Usability: Definition Usability

4 Vgl. Herczeg 2009, S. 7

5 DIN EN ISO 9241 Teil 11, S. 4

6 Vgl. Böhringer 2008, S. 466

7 Vgl. Dahm 2006, S. 145

Im Rahmen dieser Arbeit sind besonders die Regeln für Webseiten und Anwendungen wichtig. Aus diesem Grund werden besonders die Teile 11 (Anforderungen an die Gebrauchstauglichkeit; Leitsätze) und 110 (Grundsätze der Dialoggestaltung) der Norm behandelt. Teil 11 der Norm gibt zunächst einen Überblick über die Gebrauchstauglichkeit und definiert in diesem Zusammenhang wichtige Begriffe wie Effektivität und Effizienz. Diese Ausführungen bilden die Grundlage für alle weiteren Betrachtungen. Konkrete Kriterien die auf Webseiten und Anwendung bezogen werden können, finden sich in Teil 110 der Norm. Hier werden wichtige Eigenschaften vorgestellt, die sowohl bei der Evaluation als auch der Entwicklung von Software Beachtung finden sollten.

Weiterhin steht bei der Erstellung von Webseiten und Anwendungen die Darstellung von Informationen im Vordergrund. Teil 12 der Norm ISO 9241 behandelt dieses Thema und wird daher ebenfalls in den folgenden Kapiteln vorgestellt.

2.1.1.2 ISO 9241-11: Anforderungen an die Gebrauchstauglichkeit

Die Norm ISO 9241-11 enthält neben einer Definition der Begriffe Gebrauchstauglichkeit, Effektivität, Effizienz und Zufriedenstellung Erläuterungen zum Nutzungskontext. In Abbildung 1 sind die einzelnen Bestandteile der Norm und deren Zusammenhänge zu sehen.

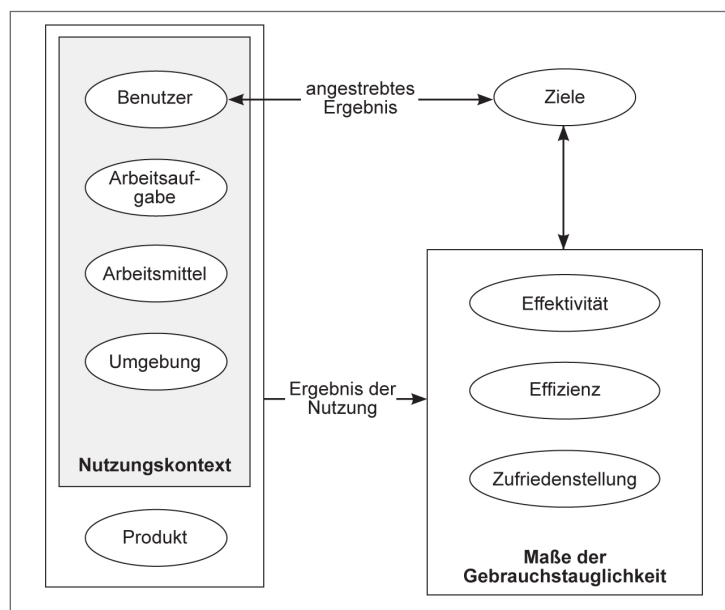


Abbildung 1 Übersicht über ISO 9241-11⁸

Das Produkt muss immer in einem Nutzungskontext gesehen werden. Dieser Kontext besteht aus dem Benutzer, der Arbeitsaufgabe, den Arbeitsmitteln und der Arbeitsumgebung. Der Zusammenhang wird auch in Abbildung 1 deutlich. Diese Komponenten haben einen direkten Einfluss auf die Effektivität, Effizienz und Zufriedenstellung bei der Benutzung.

Die Effektivität beschreibt „die Genauigkeit und Vollständigkeit mit der Benutzer ein bestimmtes Ziel erreichen“⁹. Die vollständige und korrekte Lösung einer Aufgabe ist somit eines der wichtigsten Anliegen bei der Nutzung eines interaktiven Systems.

Setzt man die Genauigkeit und Vollständigkeit des erreichten Ziels ins Verhältnis zum Aufwand, bildet dies den Leitsatz Effizienz. Der Aufwand kann für den Benutzer von physischer oder psychischer Art sein sowie Zeit, Material und Kosten betreffen. Allgemein sollte der Aufwand zum Erreichen von Zielen immer so gering wie möglich sein.

Der dritte Leitsatz beschäftigt sich mit der Zufriedenstellung des Benutzers. Diese subjektive Eigenschaft beschreibt „das Ausmaß, in dem Benutzer von Beeinträchtigungen frei sind und ihre Einstellungen zur Nutzung des Produktes“.¹⁰

2.1.1.3 ISO 9241-110: Grundsätze der Dialoggestaltung

Die Richtlinien der Norm ISO 9241-110 können für alle interaktiven Systeme in jedem beliebigen Nutzungskontext verwendet werden, da sie allgemeingültig und unabhängig von Arbeitsaufgabe, Umgebung und verwendeter Technik sind. Daher sind diese auch für den mobilen Kontext geeignet und bilden eine wichtige Grundlage für aufbauende Richtlinien.

Die Norm ISO 9241-110 beinhaltet sieben Kriterien, welche die Interaktion zwischen Benutzer und interaktivem System behandeln. Bei diesen Kriterien handelt es sich um Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit, Erwartungskonformität, Lernförderlichkeit, Steuerbarkeit, Fehlertoleranz und Individualisierbarkeit.

Aufgabenangemessenheit

„Ein interaktives System ist aufgabenangemessen, wenn es den Benutzer unterstützt, seine Arbeitsaufgabe zu erledigen, d.h., wenn Funktionalität und Dialog auf den charakteristischen Eigenschaften der Arbeitsaufgabe basieren, anstatt auf der zur Aufgabenerledigung eingesetzten Technologie.“¹¹

Das Kriterium Aufgabenangemessenheit bezieht sich direkt auf die Effektivität und Effizienz im Bezug auf die Arbeitsaufgabe des Benutzers.¹² Diese Aufgabe sollte ohne Behinderungen oder Ablenkungen schnell und erfolgreich abgeschlossen werden können. Dafür sind nur relevante Informationen anzuzeigen. Zusatzinformationen sollten ausschließlich auf Anforderung des Benutzers, zum Beispiel in einer Hilfe, verfügbar sein.¹³

Die folgende Auflistung fasst die Grundsätze der Aufgabenangemessenheit entsprechend der Norm ISO 9241-110 zusammen.

9 DIN EN ISO 9241 Teil 11, S. 4

10 DIN EN ISO 9241 Teil 11, S. 7

11 DIN EN ISO 9241 Teil 110, S. 8

12 Vgl. Herczeg 2009, S. 169

13 Vgl. DIN EN ISO 9241 Teil 110, S. 9

- Nur Informationen anzeigen, die für die Erledigung der Arbeitsaufgabe notwendig sind
- Form der Eingabe und Ausgabe sollte der Arbeitsaufgabe entsprechen
- Typische Eingabewerte sollten als voreingestellte Werte verfügbar sein
- Dialogschritte sollten dem Arbeitsablauf entsprechen
- Ein- und Ausgabemedien sollten der Aufgabe entsprechen

Selbstbeschreibungsfähigkeit

„Ein Dialog ist in dem Maße selbstbeschreibungsfähig, in dem für den Benutzer zu jeder Zeit offensichtlich ist, in welchem Dialog, an welcher Stelle im Dialog er sich befindet, welche Handlungen unternommen werden können und wie diese ausgeführt werden können.“¹⁴

Die Benutzer sollten sich grundsätzlich ohne Erklärungen oder Hilfestellung in einem Dialog zurechtfinden. Dafür sind Anleitungen und Informationen anzuzeigen, die den Benutzer bei der Bearbeitung seiner Aufgabe unterstützen. Weiterhin sollten Rückmeldungen und Zustandsinformationen bereitgestellt werden, um dem Benutzer den Zustand des Systems zu kommunizieren. Weiterhin müssen alle Interaktionsmöglichkeiten, die man als Nutzer hat, offensichtlich sein. Diese und weitere Punkte der Richtlinie Selbstbeschreibungsfähigkeit sind in der folgenden Liste aufgeführt.

- Informationen (Anleitungen, Rückmeldungen, Zustandsinformationen) anzeigen um den Benutzer erfolgreich durch den Dialog zu leiten
- Notwendigkeit von Benutzerhandbüchern und externen Hilfsmitteln minimieren
- Über Änderungen des Zustandes informieren (erwartete Eingaben, Überblick über folgende Dialogschritte)
- Informationen über die erwartete Eingabe bereitstellen
- Interaktionen sollten für den Nutzer offensichtlich sein
- Erwartete Formate und Einheiten bereitstellen

Erwartungskonformität

„Ein Dialog ist erwartungskonform, wenn er den aus dem Nutzungskontext heraus vorhersehbaren Benutzerbelangen sowie allgemein anerkannten Konventionen entspricht.“¹⁵

Grundsätzlich sollten Benutzeroberflächen ohne Vorwissen oder Erläuterungen verständlich und somit sofort bedienbar sein. Um dies zu ermöglichen muss auf die Erfahrungen der Benutzer zurückgegriffen werden. Solche Erfahrungen können sowohl von der Nutzung anderer Systeme stammen, aber auch aus dem Alltag. Besonders wichtig ist aus diesem Grund die Konsistenz. Diese kann in innere, äußere und metaphorische Konsistenz

14 DIN EN ISO 9241 Teil 110, S. 10

15 DIN EN ISO 9241 Teil 110, S. 11

unterteilt werden.¹⁶ Sind Dialoge innerhalb der Anwendung gleich oder ähnlich aufgebaut und gestaltet, handelt es sich um die innere Konsistenz. Treffen diese Anforderungen übergreifend auf andere Anwendungen zu, spricht man von äußerer Konsistenz. Die metaphorische Konsistenz beschreibt dagegen Vergleiche mit der realen, physikalischen Umgebung des Benutzers.¹⁷

In der folgenden Liste sind die wichtigsten Richtlinien des Kriteriums Erwartungskonformität zusammengefasst.

- Aufgrund der Arbeitsaufgabe vertrautes Vokabular nutzen
- Unmittelbare und passende Rückmeldung auf Interaktionen anbieten
- Abweichende Antwortzeiten kommunizieren
- Strukturierung und Organisation von Informationen sollten als natürlich empfunden werden
- Kulturelle und sprachliche Konventionen beachten
- Konsistenz von Dialogverhalten und Informationsdarstellung im interaktiven System und darüber hinaus sicherstellen
- Objektive und konstruktive Rückmeldungen und Informationen bereitstellen

Lernförderlichkeit

„Ein Dialog ist lernförderlich, wenn er den Benutzer beim Erlernen der Nutzung des interaktiven Systems unterstützt und anleitet.“¹⁸

Der Benutzer sollte beim Erlernen eines Systems optimal unterstützt werden. Daher sind wichtige Informationen und Konzepte zur Verfügung zu stellen. Um den Benutzer aber nicht von seiner Aufgabe abzulenken, sollten diese zusätzlichen Informationen nur auf Anforderung bereitgestellt werden.

Es kann hilfreich sein, die Benutzer entsprechend ihrer Erfahrungen in verschiedene Gruppen einzuteilen.¹⁹ Durch die Unterscheidung zum Beispiel in Einsteiger und Fortgeschrittener können entsprechend ausführliche oder reduzierte Informationen angeboten werden.

Weitere wichtige Regeln zur Lernförderlichkeit sind in folgender Übersicht zusammengefasst.

- Wichtige Regeln und Konzepte zum Erlernen des Systems zur Verfügung stellen
- Unterstützung beim ersten und erneuten Erlernen anbieten
- Rückmeldungen und Erläuterungen für konzeptionelles Verständnis bereitstellen
- Rückgängig machen und Wiederherstellen ermöglichen
- Minimaler Lernaufwand, zusätzliche Informationen auf Anfrage

¹⁶ Vgl. Herczeg 2009, S. 177

¹⁷ Vgl. Herczeg 2009, S. 177

¹⁸ DIN EN ISO 9241 Teil 110, S. 12

¹⁹ Vgl. Herczeg 2009, S. 178

Steuerbarkeit

„Ein Dialog ist steuerbar, wenn der Benutzer in der Lage ist, den Dialogablauf zu starten sowie seine Richtung und Geschwindigkeit zu beeinflussen, bis das Ziel erreicht ist.“²⁰

Dialoge sollten grundsätzlich immer durch den Benutzer gesteuert werden und nie durch das System. Auf diese Weise kann der Benutzer selbst über Folgeschritte und über die Arbeitsgeschwindigkeit bestimmen. Ist eine Zeitbeschränkung oder eine Steuerung durch das System für die Arbeitsaufgabe notwendig, muss die Richtlinie entsprechend angepasst werden.

Die Steuerbarkeit von Dialogen schließt auch das Zurücksetzen von Interaktionen mit ein. Diese Möglichkeit sollte unterstützt werden, falls es für die Arbeitsaufgabe sinnvoll ist.

Eine Zusammenfassung des Kriteriums Steuerbarkeit findet sich in folgender Übersicht.

- Benutzer bestimmt Geschwindigkeit von Interaktionen
- Benutzer steuert Fortsetzung des Dialogs
- Unterbrochener Dialog kann an beliebiger Stelle wieder aufgenommen werden
- Dialogschritte können rückgängig gemacht werden
- Umfang von großen Datenmengen steuerbar machen
- Jedes verfügbare Ein- und Ausgabegerät nutzbar wenn möglich
- Voreingestellte Werte sind änderbar, Originaldaten können wieder hergestellt werden

Fehlertoleranz

„Ein Dialog ist fehlertolerant, wenn das beabsichtigte Arbeitsergebnis trotz erkennbar fehlerhafter Eingaben entweder mit keinem oder mit minimalem Korrekturaufwand seitens des Benutzers erreicht werden kann. Fehlertoleranz wird mit den Mitteln erreicht: Fehlererkennung und Fehlervermeidung (Schadensbegrenzung); Fehlerkorrektur und Fehlermanagement, um mit Fehlern umzugehen, die sich ereignen.“²¹

Im Idealfall werden Fehler gänzlich durch Voreinstellungen und die Vorgabe sinnvoller Werte vermieden. Da dies aber nicht immer realisiert werden kann, muss die Fehlerkennzeichnung eindeutig und die Fehlerbehebung für den Benutzer einfach und schnell sein. Wichtig ist eine verständliche und konstruktive Fehlermeldung, die den Nutzer nicht nur bei der Korrektur unterstützt, sondern auch die Lernförderlichkeit berücksichtigt.

In der folgenden Übersicht sind die Regeln für die Fehlertoleranz zusammengefasst.

20 DIN EN ISO 9241 Teil 110, S. 13

21 DIN EN ISO 9241 Teil 110, S. 14

- Eingabefehler entdecken und vermeiden helfen
- Undefinierbare Systemzustände oder Abbrüche vermeiden
- Erläuterung und Unterstützung für Beseitigung des Fehlers anbieten
- Automatische Korrektur anbieten, Möglichkeit für den Benutzer einzugreifen
- Fehlerkorrektur wenn möglich zurückstellen
- Notwendige Schritte zur Fehlerbehebung minimieren
- Für schwerwiegende Änderungen Bestätigung einholen und Erläuterungen anbieten

Individualisierbarkeit

„Ein Dialog ist individualisierbar, wenn Benutzer die Mensch-System-Interaktion und die Darstellung von Informationen ändern können, um diese an ihre individuellen Fähigkeiten und Bedürfnisse anzupassen.“²²

Das Kriterium Individualisierbarkeit ist stets mit Vorsicht zu betrachten. Zunächst sollten Anpassungsmöglichkeiten nie die ergonomische Gestaltung eines Dialoges ersetzen. Weiterhin sind die Nutzer nicht davor geschützt, unvorteilhafte Einstellungen zu wählen. Dadurch können die Gebrauchstauglichkeit von Systemen erheblich herab gesetzt und die Nutzer sogar geschädigt werden. Wichtig ist an dieser Stelle das Eingrenzen der Anpassungen, um ungewollte Beeinträchtigungen zu vermeiden.²³

Für Webseiten und Webanwendungen ist zu beachten, dass viele dieser Individualisierungen über den Browser vorgenommen werden können. Dies ist zum Beispiel die Anpassung der Schriftgröße.

- System sollte an die Eigenschaften des Benutzers anpassbar sein
- Auswahl verschiedener Formen der Informationsdarstellung ermöglichen
- Umfang von Erläuterungen und Rückmeldungen sollte anpassbar sein
- Geschwindigkeit von Ein- und Ausgaben sollte anpassbar sein
- Auswahl unterschiedlicher Dialogarten anbieten
- Dialogelemente und Funktionen können hinzugefügt und neu angeordnet werden
- Zurücksetzen der individuellen Einstellungen ermöglichen

2.1.1.4 ISO 9241-12: Informationsdarstellung

Der Benutzer sollte die für ihn relevanten Informationen effektiv, effizient und zufriedenstellend auffinden und verstehen können. Um diesen Anspruch zu erfüllen sollte die Gestaltung von Informationen die Eigenschaften besitzen, die in Tabelle 1 zusammengefasst sind.

²² DIN EN ISO 9241 Teil 110, S. 15

²³ Vgl. DIN EN ISO 9241 Teil 110, S. 16

Klarheit	Schnelle und genaue Vermittlung des Informationsgehalts
Unterscheidbarkeit	Angezeigte Information kann klar unterschieden werden
Kompaktheit	Beachtung der Aufgabenangemessenheit, nur notwendige Informationen anzeigen
Konsistenz	Gleiche Informationen werden auf gleiche Art und Weise dargestellt
Erkennbarkeit	Lenkung der Aufmerksamkeit auf wichtige Informationen
Lesbarkeit	Informationen sind leicht zu lesen
Verständlichkeit	Bedeutung von Informationen ist leicht verständlich, eindeutig und interpretierbar

Tabelle 1 Eigenschaften von Informationen²⁴

Organisation von Informationen

Damit die Nutzer wichtige Informationen schnell aufnehmen und verstehen können, müssen diese geordnet und voneinander abgegrenzt platziert und dargestellt werden. In Tabelle 2 sind wichtige Richtlinien aufgeführt, unter anderem für die Gruppierung von Informationen und für die Beschriftung.

Platzierung	Erwartungskonformität und Aufgabenangemessenheit beachten
Fenster	Eindeutige Identifikation durch einen Titel Konsistente Gestaltung der Fenster
Bereiche	Konsistente Position Menge an Informationen beschränken
Gruppierung	Unterscheidbarkeit durch Lage und Abstand Reihenfolge entsprechend der Arbeitsaufgabe Anzahl der Elemente für einen schnellen Überblick reduzieren
Listen	Reihenfolge logisch, entsprechend der Aufgabe Visuelle Trennung für Strukturierung
Beschriftungen	Elemente sollten immer beschriftet sein Beschriftung in der Nähe des Elementes Format und Ausrichtung konsistent
Formularfelder	Eingabe- und Anzeigefelder unterscheidbar Unterschied von Nutzereingaben und voreingestellten Werten Erwartete Formate angeben

Tabelle 2 Organisation von Informationen²⁵

²⁴ Vgl. DIN EN ISO 9241 Teil 12, S. 7

²⁵ Vgl. DIN EN ISO 9241 Teil 12, S. 7ff

Kodierung von Informationen

Um Informationen richtig zu kodieren, sollte sich an bestehenden Konventionen und an den Erwartungen der Benutzer orientiert werden. So sind zum Beispiel Fehler in einem System mit einem Warndreieck und in roter Farbe gestaltet. Erfolgsmeldungen hingegen grün und mit einem Häkchen.

In der Übersicht in Tabelle 3 sind die wichtigsten Richtlinien für die gebrauchstaugliche Nutzung von Codes aufgeführt.

Allgemein	Unterscheidbarkeit der Codes Konsistente und erwartungskonforme Gestaltung
Grafische Kodierung	Symbole / Icons eindeutig, klar und verständlich
Kodierung mittels Farbe	Farbe sollte nie der einzige Code sein Pro Informationskategorie nur eine Farbe Erwartungskonforme Verwendung Ausreichend Kontrast für Unterscheidbarkeit
Sonstige Auszeichnungen	Größe: 2-3 Größen können leicht unterschieden werden Unterstreichungen verwenden

Tabelle 3 Kodierung von Informationen²⁶

2.1.2 Mobile Usability

Grundsätzlich existiert die Empfehlung, das Design für mobile Endgeräte so einfach wie möglich zu halten. Das folgende Zitat spiegelt diese Anforderung an Einfachheit wider. Weiterhin wird deutlich, dass ein einfaches und reduziertes Design genauso ansprechend sein sollte, wie ein umfangreiches Layout.

„The best advice I can give you about mobile web design is: Keep It Simple! However, that doesn't mean Keep It Ugly.“²⁷

Diese Anforderung ergibt sich nicht nur aus der geringeren Größe der Geräte, sondern auch aus dem abweichenden Nutzungskontext. Aus diesem Grund sind die Richtlinien, die im vorherigen Kapitel vorgestellt wurden zwar als Grundlage verwendbar, müssen aber entsprechend ergänzt werden. Der Nutzungskontext, der für alle weiteren Betrachtungen eine Basis bildet, wird im folgenden Kapitel erläutert. Anschließend werden für das mobile Webdesign relevante Richtlinien aufgeführt.

2.1.2.1 Nutzungskontext

Für alle Produkte sind die Benutzer bzw. die Zielgruppen ein wichtiger Ansatzpunkt. Um die Regeln, die für die Gestaltung von Webseiten auf mobilen Endgeräten genutzt werden

²⁶ Vgl. DIN EN ISO 9241 Teil 12, S. 13ff

²⁷ Firtman 2010, S. 66

richtig anwenden zu können, ist die Betrachtung des Benutzers in seinem Nutzungskontext notwendig.

„To design well for mobile devices, it is essential to begin with what is most important: people.“²⁸

Die Arbeitsumgebung kann bei der Nutzung von mobilen Geräten nicht auf die gleiche Art und Weise definiert werden, da die Benutzer mobil und unabhängig von einem herkömmlichen Arbeitsplatz sind. Oft werden mobile Geräte nur neben einer anderen Tätigkeit verwendet, zum Beispiel beim Fahren, im Bus, während der Werbung oder während der Arbeit.²⁹

Durch die Verwendung der Geräte nebenbei ist der Benutzer abgelenkt und konzentriert sich nicht auf die Nutzung des Gerätes.³⁰ Meist ist die Nutzung sogar hektisch, da aufgrund der Umgebung oder der eigentlichen Hauptaufgabe wenig Zeit bleibt.³¹

Daher ist ein einfaches und intuitives Design besonders wichtig. Buchstaben, Buttons und Grafiken sollten entsprechend klar, groß und gut lesbar bzw. erkennbar sein. Wichtige Einstellungen müssen in Millisekunden durch den Benutzer erfasst werden können.³²

Da mobile Geräte überall verwendet werden können, sind auch die Lichtverhältnisse zu bedenken. Die Webseite oder Anwendung sollte sowohl in heller als auch dunkler Umgebung gut lesbar und nutzbar sein.³³ Weitere Störquellen oder Ablenkungen sind zum Beispiel akustischer Art oder auch schlechter Empfang.³⁴

Neben einer Nutzung von mobilen Geräten in Eile, unterwegs und in Bewegung, verwenden die Benutzer diese aber auch entspannt und in Ruhe zu Hause. Es ist praktisch, die handlichen Geräte immer schnell griffbereit zu haben. Die Verwendung in der Küche, im Bad oder auf dem Sofa ist keine Seltenheit.³⁵ Für eine wachsende Anzahl an Benutzer ist das mobile Endgerät sogar der einzige Weg, wie diese privat das Internet verwenden.³⁶

Grundsätzlich wollen die Nutzer bei der Verwendung von mobilen Geräten zu Hause nicht viel Energie aufbringen und sich nicht anstrengen.³⁷ Auch in diesem Nutzungskontext muss die Verwendung daher einfach und intuitiv sein.

Aufgrund des sehr gegenseitigen Nutzungskontextes wird klar, dass dieser an sich für die mobile Verwendung schwer zu definieren ist. Es gibt keinen festgelegten Arbeitsplatz, keine fest definierte physische und soziale Umgebung. Nur der Benutzer, also die Zielgruppe mit ihren Bedürfnissen und Fähigkeiten kann bestimmt werden. Weiterhin ist es möglich die Arbeitsaufgaben festzulegen. Auch die Nutzung im mobilen Kontext ist meist

28 Layon 2012, S. 1

29 Vgl. Layon 2012, S. 5
Vgl. Wee ; Lang: Writing Effective Mobile Web Content

30 Vgl. Layon 2012, S. 7

31 Vgl. Keck: App-Usability, Herausforderungen und Guidelines

32 Vgl. Layon 2012, S. 5ff

33 Vgl. Hoover ; Berkman: Designing Mobile Interfaces, Principles of Mobile Design

34 Vgl. Keck: App-Usability, Herausforderungen und Guidelines

35 Vgl. Layon 2012, S. 8
Vgl. Clark: Nielsen is wrong on mobile, 12.4.2012

36 Vgl. Clark: Nielsen is wrong on mobile, 12.4.2012

37 Vgl. Layon 2012, S. 8f

aufgabenorientiert. Egal ob der Benutzer unterwegs oder zu Hause ist, er möchte mit Hilfe des mobilen Gerätes Aufgaben ausführen. Diese müssen ebenfalls effektiv, effizient und zufriedenstellend erledigt werden können.³⁸

2.1.2.2 Konsistenz

Bereits in der Norm ISO 9241-110 wird die Erwartungskonformität und somit die Konsistenz außerhalb und innerhalb der Webseite bzw. Anwendung hervorgehoben. Dies gilt auch in Bezug auf die mobile Ansicht der Webseite. Diese sollte unbedingt konsistent zur Desktop-Webseite sein.³⁹ Zumindest die Hauptelemente sollten nicht geändert oder weggelassen werden. Die Hauptnavigation, die Struktur der Webseite, Farben und Grafiken müssen ähnlich sein, um dem Benutzer ein Wiedererkennen zu ermöglichen.⁴⁰

Dieser Anspruch unterstützt auch das Konzept „One Web“, denn dem Benutzer wird die gleiche Webseite geliefert, nur Darstellung und Funktionsumfang sind für die mobilen Geräte angepasst.⁴¹

2.1.2.3 Navigation

Da der sichtbare Bereich einer Webseite oder Anwendung auf einem mobilen Gerät sehr gering ist und der Benutzer bereits ohne Scrollen den Inhalt sehen sollte, muss der Umfang der primären Navigation im Kopfbereich reduziert werden.⁴² Diese Hauptnavigation sollte aus maximal drei bis vier Links bestehen, damit die Übersichtlichkeit erhalten bleibt und der Platz gut ausgenutzt werden kann.⁴³ Falls eine umfangreiche Navigation notwendig ist, können die entsprechenden Links unterhalb des Inhalts im Footer angezeigt werden.⁴⁴ Um dem Benutzer vom Footer aus einen schnellen Weg zurück zur Hauptnavigation anzubieten, sollte ein „Nach oben“ Link im Footer hinterlegt werden.⁴⁵

Eine Alternative ist das Fixieren der Links am unteren Bildschirmrand wie in Abbildung 2 dargestellt. Auf diese Weise ist die Navigation für den Nutzer immer zu sehen und an der gleichen Stelle zu finden. Ein weiterer Vorteil liegt in der besseren Bedienbarkeit, da der untere Bereich des Gerätes näher an den Fingern des Benutzers liegt.⁴⁶ Außerdem ist die unten fixierte Navigation vermehrt auch bei nativen Apps zu finden, sodass der Benutzer an dieser Stelle vorhandene Erfahrungen und Kenntnisse nutzen kann. In Abbildung 2 ist eine solche native App unter iOS zu sehen.

38 Vgl. Layon 2012, S. 10

39 Vgl. Firtman 2010, S. 66

40 Vgl. Mobile Moxie: 16 Mobile Site Usability Mistakes to Avoid, 13.2.2012
Vgl. Hooper ; Berkman: Designing Mobile Interfaces, Page

41 Vgl. W3C: Mobile Web Best Practices 1.0, Basic Guidelines

42 Vgl. W3C: Mobile Web Best Practices 1.0, Basic Guidelines

43 Vgl. Firtman 2010, S. 67

44 Vgl. W3C: Mobile Web Best Practices 1.0, Basic Guidelines

45 Vgl. Firtman 2010, S. 67

46 Vgl. Firtman 2010, S. 70

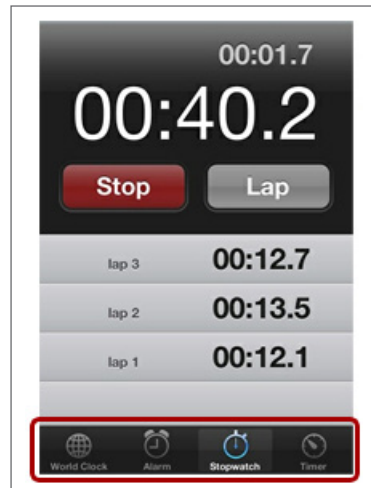


Abbildung 2 Hauptnavigation am unteren Bildschirmrand

Auch im mobilen Kontext ist die Orientierung des Nutzers sehr wichtig. Daher sollte die Navigation innerhalb der gesamten Anwendung konsistent dargestellt sein.⁴⁷ Grundsätzlich sollte die Anzahl der notwendigen Navigationsschritte auf maximal vier reduziert werden, da der Benutzer schnell und mit wenigen Seitenaufrufen zu seinem Ziel kommen muss. Die wichtigsten Informationen und die häufig genutzten Funktionen sollten daher mit wenigen Schritten erreichbar sein.⁴⁸

2.1.2.4 Informationsdarstellung

Besonders bei mobilen Versionen von Webseiten und Anwendungen steht der Inhalt im Vordergrund. Daher sollten sonstige Elemente, wie der Kopfbereich oder die Navigation nicht zu viel Platz weg nehmen und Informationen sofort sichtbar und zugänglich sein.⁴⁹ Die wichtigsten Inhalte und Funktionen sind aus diesem Grund oben zu platzieren, sodass der Benutzer diese ohne Scrollen sofort sehen kann.⁵⁰

Weiterhin sollte die gesamte Breite des Gerätes für den Inhalt genutzt werden. Mehrere Spalten sind aufgrund der geringen Größe nicht zu empfehlen.⁵¹ Eine Ausnahme bilden an dieser Stelle Tablets, denn mit diesen Geräten steht wesentlich mehr Platz für den Inhalt zur Verfügung.

Besonders bei auswählbaren Elementen wie Links oder Listen-Einträgen ist das Ausnutzen der gesamten Breite wichtig.⁵² Der Benutzer selektiert diese mit den Fingern, sodass die Elemente eine bestimmte Größe haben müssen. Geht das Element über die gesamte Breite

47 Vgl. W3C: Mobile Web Best Practices 1.0, Basic Guidelines

48 Vgl. W3C: Mobile Web Best Practices 1.0, Basic Guidelines

49 Vgl. W3C: Mobile Web Best Practices 1.0, Basic Guidelines

50 Vgl. Firtman 2010, S. 67

Vgl. W3C: Mobile Web Best Practices 1.0, Basic Guidelines

Vgl. Apple: iOS Human Interface Guidelines, User Experience Guidelines

51 Vgl. Firtman 2010, S. 67

52 Vgl. Firtman 2010, S. 67

des Bildschirms ist es für den Nutzer sehr viel leichter, dieses auszuwählen. Damit ist die Nutzung einfacher und Eingabefehler werden vermieden.

Die Anzahl der Kontrollelemente, Links und Listenelemente sollte stets minimal sein, damit der Nutzer die Hauptfunktionen klar erkennen und verwenden kann. Auf diese Weise werden ebenfalls Eingabefehler vermieden und die Nutzung wird klarer und intuitiver.⁵³

Die tabellarische Darstellung von Informationen ist für große Datensätze im Web weit verbreitet. Für eine mobile Webseite sollten allerdings eher Listen statt Tabellen genutzt werden.⁵⁴ Um umfangreiche Tabellen lesen zu können, müssen diese entsprechend vergrößert werden. Auf diese Weise ist es dem Benutzer allerdings nicht möglich einen Zusammenhang zwischen den Kopfzeilen und dem Inhalt herzustellen, da nur ein Ausschnitt der Tabelle auf dem Bildschirm sichtbar ist.

Wichtig bei der Verwendung von Listen ist eine sinnvolle Gruppierung der Einträge damit die Inhalte für den Benutzer übersichtlich bleiben. Eine Gruppe sollte dabei nicht mehr als zehn Einträge enthalten.⁵⁵

Weiterhin sind Overlay-Fenster im Web sehr verbreitet. Sie stellen eine gute Lösung dar, um zusätzlichen Inhalt auf Anforderung des Benutzers anzuzeigen. Auf einem mobilen Endgerät sind Fenster allerdings ungeeignet und die Verwendung sollte vermieden werden.⁵⁶ Durch den kleinen Bildschirm ist das Overlay-Fenster unter Umständen nicht vollständig zu sehen oder gar nicht im sichtbaren Bereich. Der Benutzer wird dadurch verunsichert und bei der Nutzung der Anwendung unterbrochen oder behindert.⁵⁷ Im schlimmsten Fall unterstützt das jeweilige Gerät die Nutzung von mehreren Fenstern nicht und die Seite wird völlig unbenutzbar.⁵⁸ Das Platzieren der Inhalte auf einer zusätzlichen Seite, auf die der Nutzer über einen Link gelangt, kann eine sinnvolle Alternative sein.

Um den Benutzer nicht zu irritieren sollte weiterhin kein automatisches Neu-Laden der Seite stattfinden. Eine Ausnahme ist es, wenn der Nutzer darüber informiert und damit einverstanden ist. Problematisch ist das Neu-Laden der Seite vor allem bei langsamer oder teurer Internetverbindung.⁵⁹

2.1.2.5 Interaktionen

Geräte mit Touchscreen werden mit Hilfe direkter Manipulation bedient, d.h. die Steuerung erfolgt über Gesten, aber auch über die Bewegung des gesamten Gerätes.⁶⁰ Es existieren bereits zahlreiche Standard-Gesten, die genutzt werden sollten. Diese sind in Tabelle 4 dargestellt. Von neuen, individuellen Gesten sollte wenn möglich abgesehen werden, da die Benutzer hier keine Erfahrungen haben und die neue Funktion erst erlernen müssen.⁶¹

53 Vgl. Apple: iOS Human Interface Guidelines, User Experience Guidelines

54 Vgl. Firtman 2010, S. 67

55 Vgl. Firtman 2010, S. 67

56 Vgl. W3C: Mobile Web Best Practices 1.0, Basic Guidelines
Vgl. Seward 2011, S. 6

57 Vgl. W3C: Mobile Web Best Practices 1.0, Basic Guidelines

58 Vgl. Seward 2011, S. 6

59 Vgl. W3C: Mobile Web Best Practices 1.0, Basic Guidelines

60 Vgl. Apple: iOS Human Interface Guidelines, Human Interface Principles

61 Vgl. Apple: iOS Human Interface Guidelines, User Experience Guidelines

Schwierige Gesten sollten nur als Ergänzung zum herkömmlichen Weg über einen Button genutzt werden. An dieser Stelle werden zwei bis drei zusätzliche Schritte in Kauf genommen, dafür ist die Funktion für den Benutzer aber sofort präsent. Er hat aber trotzdem die Option, die schwierige Geste zu nutzen.⁶²








Geste		Beschreibung
Tap (Antippen)		Öffnen Aktivieren Selektieren Stoppen des Scrollens Analog zum Klick mit der linken Maustaste
Drag / Swipe (Ziehen)		Objekt bewegen Scrollen oder Blättern
Flick / Fling, Slide (Schnelles Wischen)		Schnelles Bewegen von Objekten Scrollen iOS: schnelles Scrollen und Klicken
Double Tap (Doppelt Antippen)		iOS: Zoom in / out, Bereich zentrieren Android: Zoom in, Text selektieren Windows: Zoom
Pinch (Zusammenziehen)		Verkleinern
Spread (Auseinanderziehen)		Vergrößern
Touch and hold (Antippen und halten)		iOS: vergrößerte Ansicht zum exakten Positionieren des Cursors (bei Text) Android: Auswählen von mehreren Dateien (Sammeloperation) Windows: Öffnen eines Kontextmenüs

Tabelle 4 Gesten für die Bedienung eines Touchscreens⁶³

Infolge der Bedienung mit Gesten werden alle Interaktionen mit den Fingern durchgeführt. Daher muss der aktive bzw. auswählbare Bereich entsprechend größer sein, als bei der Verwendung einer Maus. Als Minimum wird die Fläche von 40 x 40 Pixel für wichtige und oft

62 Vgl. Apple: iOS Human Interface Guidelines, User Experience Guidelines

63 Saffer 2008, S. 181f

Vgl. Apple: iOS Human Interface Guidelines, Platform Characteristics

Vgl. Android Developer: Android Design, Gestures

zu nutzende Buttons und Links empfohlen. Für selten benutzte Elemente kann die Größe auch auf 25 Pixel reduziert werden.⁶⁴

Apple empfiehlt als minimale Größe 44 x 44 Punkt, Microsoft 38 x 38 Pixel.⁶⁵ Aus den unterschiedlichen Maßeinheiten wird deutlich, dass es stets auf die Auflösung des Gerätes ankommt, welche Pixelgrößen oder Punktmaße genutzt werden sollten. Die tatsächlich nutzbare Größe für den Benutzer sollte nicht unter 1 x 1 Zentimeter liegen.⁶⁶ Je höher die Auflösung ist, desto größer müssen auch die auswählbaren Elemente gestaltet werden. Hier empfiehlt sich stets ein Test direkt auf den entsprechenden Geräten.⁶⁷

Auch der Abstand zwischen auswählbaren Elementen sollte an die Interaktion mit dem Finger angepasst werden. Empfohlen werden 20 Pixel oder mehr Zwischenraum, um ein eindeutiges Auswählen einer Schaltfläche zu ermöglichen.⁶⁸

Im Bereich der mobilen Webseite kann auch ein sogenannter „iceberg tip“, wie in Abbildung 3 zu sehen, genutzt werden. Diese haben einen größeren aktiven Bereich (gestrichelte Linie), der aber nicht sichtbar ist. Auf diese Weise bleibt das Layout erhalten, der Benutzer kann aber trotzdem sicher mit den Fingern auswählen. Ein Abstand zum aktiven Bereich des nächsten Elements muss entsprechend gegeben sein.⁶⁹



Abbildung 3 Iceberg Tip⁷⁰

Ist die richtige Größe für Interaktionselemente sichergestellt, müssen diese auch entsprechend gestaltet sein, sodass der Nutzer die Interaktionsmöglichkeit entdeckt und versteht. Diese so genannten Call-To-Action Elemente sollten noch deutlicher als für den Desktop-Kontext als solche zu erkennen sein.⁷¹ Abbildung 4 zeigt ein gutes Beispiel für die Gestaltung von Interaktionselementen. Unter iOS ist der Anrufen-Button entsprechend gestaltet und erregt die Aufmerksamkeit des Benutzers durch die grüne Farbe. Der rote Button hingegen warnt durch seine Farbe vor einer gegebenenfalls kritischen Aktion. Weiterhin sollten die

64 Vgl. Firtman 2010, S. 70

65 Vgl. Apple: iOS Human Interface Guidelines, User Experience Guidelines
Vgl. Microsoft: Designing Applications for Windows Mobile Platforms, Accessibility and Ergonomic Guidelines, 19.4.2012

66 Vgl. Saffer 2008, S. 42

67 Vgl. Saffer 2008, S. 42

68 Vgl. Firtman 2010, S. 70

69 Vgl. Saffer 2008, S. 43

70 Saffer 2008, S. 43

71 Vgl. Keck: App-Usability, Herausforderungen und Guidelines

Elemente in der Nähe des dazugehörigen Inhaltes platziert sein, damit der Benutzer die Beziehung erkennt und die Elemente schnell erreichen kann.⁷²



Abbildung 4 Call-To-Action unter iOS⁷³

Bei der Interaktion mit einem Touchscreen ist das unmittelbare Feedback oder das sofortige Anzeigen von Ergebnissen besonders wichtig.⁷⁴ Bei einem Tap kann der Benutzer meist sehen, dass die Schaltfläche gedrückt wird. Bei Gesten wie dem Wischen muss der Benutzer informiert werden, dass das System die Geste verstanden hat und eine Aktion ausführt.⁷⁵

Bei physischen Tasten, zum Beispiel bei der Tastatur, bekommt der Benutzer durch die Beschaffenheit der Taste an sich das notwendige Feedback. Bei einem Touchscreen ist diese Art von Feedback nicht möglich, sodass visuelles Feedback, wie zum Beispiel das Hervorheben der Taste, gegeben werden muss. Alternativ kann auch auf die Vibration oder den Ton der Geräte zurückgegriffen werden.⁷⁶

Sollte die Aktion nicht sofort ausgeführt werden können, muss dies dem Nutzer ebenfalls mitgeteilt werden. Zu diesem Zweck können Ladebalken oder Statusmeldungen genutzt werden.⁷⁷ Bei fehlender Rückmeldung des Systems wird der Nutzer unter Umständen verunsichert oder sogar frustriert oder wiederholt die Aktion mehrmals, was zu Problemen bei der Verarbeitung führen kann.⁷⁸

Eine weitere Schwierigkeit die durch die Bedienung des Touchscreens mit den Fingern entsteht, ist die Verdeckung bestimmter Bereiche mit der Hand. Dabei sind auch Linkshänder

72 Vgl. Saffer 2008, S. 42

73 Keck: App-Usability, Herausforderungen und Guidelines

74 Vgl. Firtman 2010, S. 70

Vgl. Apple: iOS Human Interface Guidelines, Human Interface Principles

75 Vgl. Saffer 2008, S. 20

76 Vgl. Keck: App-Usability, Herausforderungen und Guidelines

77 Vgl. Saffer 2008, S. 20

Vgl. Apple: iOS Human Interface Guidelines, Human Interface Principles

78 Vgl. Hooper ; Berkman: Designing Mobile Interfaces, General Interactive Controls

oder eine Bedienung mit beiden Händen gleichzeitig zu beachten.⁷⁹ In den dadurch nicht sichtbaren Bereichen sollten keine wichtigen Elemente oder Informationen platziert werden. Oft werden Buttons am unteren Bildschirmrand angeordnet, damit der Benutzer den oberen Bereich auch bei der Interaktion mit dem Gerät sehen kann.

Weiterhin ist zu beachten, dass die im Web verbreitete Mouseover- bzw. Hover-Funktion bei der Interaktion mit einem Touchscreen unter Umständen nicht möglich ist.⁸⁰ Entweder wird das Element durch einen Tap aktiviert oder es wird keine Interaktion ausgelöst. Daher sind alle Inhalte und Funktionen wie zum Beispiel Tooltips oder auch Teile der Navigation über mobile Endgeräte nicht zugänglich. Diese Informationen müssen folglich auf eine andere Art und Weise angezeigt werden. Eine Ausnahme bilden an dieser Stelle Touchscreens, die auch mit Hilfe eines Stiftes bedient werden können.

Neben dem Mouseover-Effekt ist auch die Anzeige des Kontextmenüs problematisch.⁸¹ Bei der Bedienung mit der Maus ist an dieser Stelle die rechte Maustaste verfügbar, um die zusätzlichen Inhalte aufzurufen. In einigen mobilen Betriebssystemen kann das Kontextmenü mit der Geste Touch and Hold (halten länger als 2 Sekunden) aktiviert werden.⁸²

Gerade durch den eingeschränkten Platz wird der Benutzer meist Scrollen müssen, um alle Inhalte einer Seite zu sehen.⁸³ Dabei ist horizontales Scrollen bei mobilen Endgeräten durch Funktionen wie zum Beispiel das Blättern besser benutzbar als auf einem Rechner. Grundsätzlich sollte der Benutzer nur in eine Richtung scrollen müssen, also entweder vertikal oder horizontal.⁸⁴ Wenn der Benutzer allerdings die Zoom-Funktion nutzt, muss unter Umständen in beide Richtungen gescrollt werden.⁸⁵ In diesem Fall ist der Zustand aber erwartungskonform und somit tolerierbar.

Wichtig beim Scrollen auf mobilen Endgeräten ist das Anbieten einer Orientierung, an welcher Position auf der Seite sich der Benutzer relativ zum gesamten Inhalt befindet. Dies kann durch Scrollbalken gezeigt werden. Gleichzeitig weisen diese den Benutzer darauf hin, dass weiterer Inhalt durch Scrollen verfügbar ist.⁸⁶ Es sollten allerdings keine Pfeile angezeigt werden. Diese geben den Hinweis, dass der Benutzer mit den Scrollbalken an sich interagieren kann. Bedingt durch die geringe Größe ist dies allerdings nicht möglich. Das Scrollen wird statt dessen über das Anfassen und Ziehen der gesamten Seite ausgelöst.⁸⁷

Um Platz zu sparen können die Scrollbalken auch nur angezeigt werden, wenn der Benutzer diese Aktion ausführt. Ansonsten sind die Indikatoren ausgeblendet. Apple nutzt diese Funktion zum Beispiel anwendungsübergreifend in iOS. Zusätzlich wird Feedback angeboten, wenn der scrollbare Bereich zu Ende ist.⁸⁸

79 Vgl. Firtman 2010, S. 70

80 Vgl. Saffer 2008, S. 25

81 Vgl. Saffer 2008, S. 26

82 Vgl. Firtman 2010, S. 70

83 Vgl. Saffer 2008, S. 52

84 Vgl. W3C: Mobile Web Best Practices 1.0, Basic Guidelines

85 Vgl. Hooper ; Berkman: Designing Mobile Interfaces, Scroll

86 Vgl. Hooper ; Berkman: Designing Mobile Interfaces, Scroll
Vgl. Saffer 2008, S. 52

87 Vgl. Hooper ; Berkman: Designing Mobile Interfaces, Scroll

88 Vgl. Saffer 2008, S. 52

2.1.2.6 *Formulare*

Aus Benutzersicht sollten Eingaben bei mobilen Endgeräten auf ein Minimum reduziert werden. Es wird sogar empfohlen, die freie Eingabe von Texten wenn möglich zu vermeiden.⁸⁹ Eingaben sind auf mobilen Endgeräten mit kleinen Bildschirmen und aus diesem Grund auch kleinen Tastaturen sehr mühsam und umständlich.⁹⁰ Die Fehlerrate ist durch die kleinen aktiven Bereiche der einzelnen Buchstaben relativ hoch. Meist sind die Benutzer auch in Eile und haben somit keine Zeit oder oft auch keine Lust viel Text einzugeben. Statt freien Eingaben können dem Benutzer sinnvolle Vorgaben angeboten werden, aus denen er auswählen muss. Dies kann durch Radiobuttons, Checkboxes oder Auswahllisten realisiert werden.⁹¹

Sind trotzdem Eingaben notwendig, sollte der Benutzer optimal bei dieser Arbeit unterstützt werden. Beim Anbieten von Labels und Hinweisen ist die Anordnung dieser besonders wichtig. Die Labels sollten über dem Formularfeld und die Hinweise darunter angeordnet werden. Zum einen ist bedingt durch den kleinen Bildschirm ein Positionieren nebeneinander nicht sinnvoll. Weiterhin wird bei der Auswahl eines Feldes unter Umständen heran gezoomt, damit der Benutzer seine Eingabe besser sieht. In diesem Fall sind Label und Hinweise nicht mehr im sichtbaren Bereich und während der Eingabe für den Benutzer somit nicht zugänglich.⁹²

2.1.3 *Usability nativer Apps*

Neben den Richtlinien für die Usability auf mobilen Endgeräten, die im vorherigen Kapitel vorgestellt wurden, sind weiterhin die Besonderheiten der jeweiligen Betriebssysteme zu beachten. Jeder Hersteller definiert ein bestimmtes Aussehen und Verhalten der verfügbaren Elemente. Die Benutzer sind dies gewohnt und erwarten, dass jede App auf die gleiche Weise funktioniert und die Elemente wieder erkennbar sind. In den folgenden Kapiteln werden die wichtigsten Elemente und deren Aussehen sowie deren Funktionsweise für die Systeme iOS, Android und Windows Phone beschrieben.

2.1.3.1 *Allgemein*

Für alle Betriebssysteme ist die Konsistenz das wichtigste. Die Gestaltung, Anordnung und Funktion von Elementen sollte nicht nur innerhalb der App, sondern auch im Bezug auf das

89 Vgl. W3C: Mobile Web Best Practices 1.0, Basic Guidelines

Vgl. Firtman 2010, S. 67

Vgl. Apple: iOS Human Interface Guidelines, User Experience Guidelines

90 Vgl. Hoover ; Berkman: Designing Mobile Interfaces, Principles of Mobile Design

Vgl. Seward 2011, S. 10

91 Vgl. W3C: Mobile Web Best Practices 1.0, Basic Guidelines

Vgl. Seward 2011, S. 10

92 Vgl. Firtman 2010, S. 70

System konsistent sein. Auf diese Weise ist der Benutzer sofort mit den Elementen vertraut und weiß, welche Funktion diese auslösen.⁹³

Es wird grundsätzlich empfohlen auf die Standardelemente des jeweiligen Systems zurückzugreifen und diese auch nur minimal anzupassen. Um den Benutzer optimal zu unterstützen muss die Erwartungskonformität und Wiedererkennbarkeit sichergestellt werden.⁹⁴ Daher sollten die Standardelemente des jeweiligen Systems nur für die vorgesehenen Funktionen genutzt werden. Eine abweichende Verwendung würde den Nutzer irritieren und ist nur im Ausnahmefall zu empfehlen.⁹⁵

Zu beachten ist allerdings, dass die Oberfläche jedes Betriebssystems individuell ist. Die Benutzer sind das jeweilige Aussehen und die Funktionsweise gewöhnt, sodass eine App für jedes Betriebssystem einzeln erstellt werden sollte. Auch das folgende Zitat hebt hervor, dass Benutzer eine App weniger zu schätzen wissen, die aussieht wie eine Web-App bzw. wie eine App für ein anderes Betriebssystem.

„If your app feels like it was designed for a different device, or for the web, people are less likely to value it.“⁹⁶

In Abbildung 5 ist der Unterschied zwischen iOS, Android und Windows Phone 7 zu sehen.

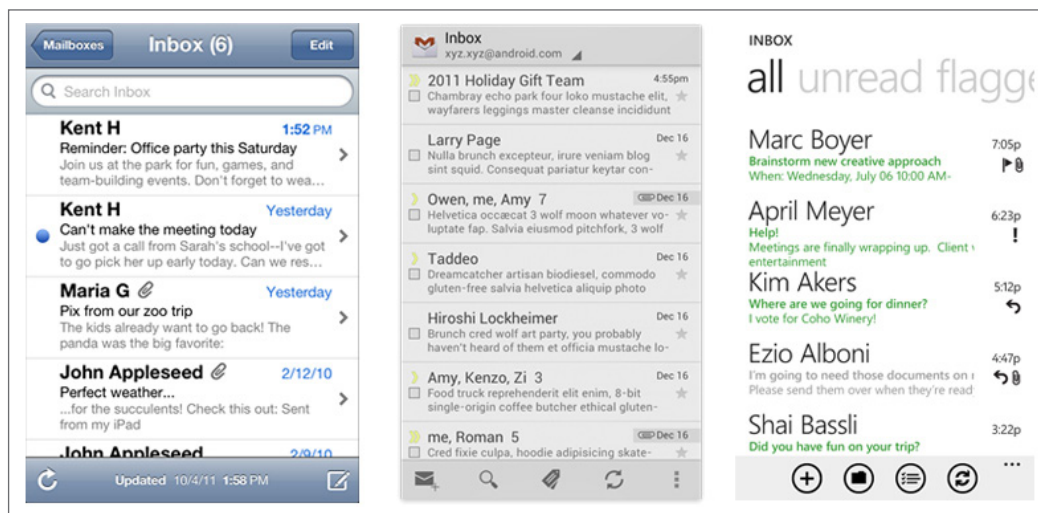


Abbildung 5 Mail Anwendung unter iOS, Android und Windows Phone 7⁹⁷

Bei nativen Apps ist es weiterhin notwendig, die Orientierung der Geräte zu berücksichtigen. Die meisten Geräte können sowohl im Hochformat (Portrait-Modus) als auch im Querformat

- 93 Vgl. Apple: iOS Human Interface Guidelines, Human Interface Principles
Vgl. Android Developer: Android Design, Design Principles
Vgl. Hoober ; Berkman: Designing Mobile Interfaces, Principles of Mobile Design
- 94 Vgl. Apple: iOS Human Interface Guidelines, App Design Strategies
Vgl. Microsoft: User Experience Design Guidelines, General Design Principles
- 95 Vgl. Apple: iOS Human Interface Guidelines, User Experience Guidelines
- 96 Apple: iOS Human Interface Guidelines, App Design Strategies
- 97 Apple: iOS Human Interface Guidelines, Case Studies: Transitioning to iOS
Android Developer: Android Design, Themes
Windows Phone: UI Design and Interaction Guide for Windows Phone 7, S. 60

(Landscape-Modus) genutzt werden. Benutzer erwarten daher, dass sich eine App immer in jeder Orientierung nutzen lässt.⁹⁸ In jeder Ausrichtung des Gerätes sollte der Inhalt im Vordergrund stehen. Das Skalieren des Inhaltes im Landscape-Modus entspricht aber nicht den Erwartungen des Nutzers. Stattdessen ist das Layout für beide Orientierungen zu optimieren.⁹⁹

Grundsätzlich sollte sich die App auch immer in der aktuellen Orientierung öffnen. Dies trifft allerdings nicht für alle Geräte zu. Das iPhone oder auch einige Geräte mit Windows Phone öffnen sich immer im Portrait-Modus, da sich der Home-Bildschirm nicht an die Position des Gerätes anpasst.¹⁰⁰

2.1.3.2 Vorgaben und Richtlinien der Hersteller

Apple stellt für sein Betriebssystem iOS, welches für den iPod Touch, das iPhone und das iPad verwendet wird, zahlreiche Gestaltungsrichtlinien zur Verfügung. In den „iOS Human Interface Guidelines“ finden sich allgemeine Regeln zur Gestaltung der Benutzeroberfläche von Apps, sowie Beschreibungen der Standardelemente und deren empfohlene Verwendung.

Apple macht in seinen Richtlinien relativ strenge Vorschriften zur Gestaltung und Funktion der Standardelemente. Dadurch wird sicher gestellt, dass diese Elemente auch noch anwendungsübergreifend zu erkennen sind. Der Benutzer kann von seinen bisherigen Erfahrungen mit den Standard-Apps unter iOS profitieren und sein Wissen daher bei neuen Apps anwenden. Der Einstieg in die Nutzung einer neuen App wird dadurch wesentlich erleichtert.

Auch Google definiert über die „Design Guidelines and Patterns“ das Aussehen und Verhalten der Benutzeroberfläche von Android. Neben allgemeinen Richtlinien werden die Standardelemente beschrieben, zu verwendende Gesten vorgestellt sowie die bevorzugte Gestaltung und Formulierung von Texten erläutert. In den Guidelines wird weiterhin betont, dass grundsätzlich nur das Android-Design zu verwenden ist und nicht zum Beispiel das von iOS kopiert werden sollte.¹⁰¹

Für das Betriebssystem Windows Phone stellt Microsoft den „UI Design and Interaction Guide“¹⁰² zur Verfügung. Die Richtlinien enthalten Empfehlungen zu den Themes, zur typographischen Gestaltung sowie zu den Standardelementen. Die möglichen Themes sind von Microsoft bereits vorgegeben. Der Benutzer muss sich für einen hellen oder dunklen Hintergrund entscheiden. Zusätzlich kann er zwischen zehn Akzent-Farben wählen. Das Theme kann überschrieben und so zum Beispiel an die Unternehmensfarben angepasst werden. Der Benutzer kann aber den Hintergrund stets anpassen. Bei der Gestaltung einer App ist also zu beachten, dass alle Elemente sowohl auf hellem als auch auf dunklem Hintergrund gut sichtbar sind. Besonders wichtig ist in diesem Zusammenhang auch die Lesbarkeit der Texte.¹⁰³

98 Vgl. Apple: iOS Human Interface Guidelines, User Experience Guidelines

99 Vgl. Apple: iOS Human Interface Guidelines, User Experience Guidelines

100 Vgl. Apple: iOS Human Interface Guidelines, User Experience Guidelines
Vgl. Windows Phone: UI Design and Interaction Guide for Windows Phone 7, S. 39

101 Vgl. Android Developer: Android Design, Pure Android

102 Windows Phone: UI Design and Interaction Guide for Windows Phone 7

103 Vgl. Windows Phone: UI Design and Interaction Guide for Windows Phone 7, S. 63

Eine weitere Besonderheit von Windows Phone ist die Verwendung von drei Orientierungen. In Abbildung 6 sind der Porträt, der Landscape-Modus links und rechts zu sehen. Der Unterschied der beiden Landscape-Modi ist die unterschiedliche Platzierung von Status und Application Bar.¹⁰⁴

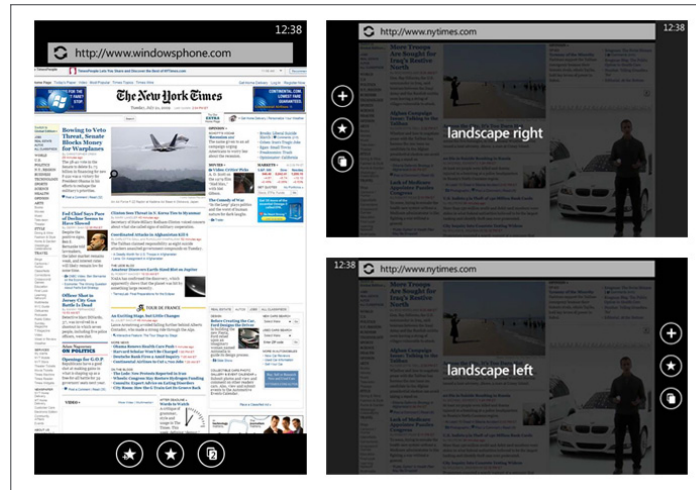


Abbildung 6 Windows Phone 7: Orientierungen¹⁰⁵

Grundsätzlich werden die gleichen Standardelemente für die Betriebssysteme iOS, Android und Windows Phone genutzt. Unterschiede gibt es vordergründig im Aussehen und bei der Verwendung. Die Elemente werden für alle Betriebssysteme im Vergleich dargestellt und Unterschiede jeweils erläutert.

Home-Bildschirm

Das Symbol der App wird bei allen Betriebssystemen auf dem Home-Bildschirm dargestellt, wie in Abbildung 7 zu sehen ist. Es sollte den Zweck des Produktes darstellen und sich von den anderen Apps möglichst abheben. Trotzdem muss es auch auf individuellen Hintergründen, zum Beispiel Fotos gut erkennbar sein.¹⁰⁶

In Abbildung 7 ist auch zu sehen, dass sich Größe und Form der Symbole in den Betriebssystemen unterscheiden. Für jedes sollte speziell ein Symbol gestaltet werden, um sich an die Vorgaben anzupassen und die bestmögliche Qualität bereit zu stellen. Weiterhin wird für iOS empfohlen, das Icon in verschiedenen Größen zu liefern, angepasst an das Gerät und dessen Auflösung. Für das iPhone und den iPod Touch sind daher Icons in den Größen 57 x 57 Pixel und 114 x 114 Pixel, für das iPad 72 x 72 Pixel sowie 144 x 114 Pixel zu erstellen. Die abgerundeten Ecken sowie Schlagschatten und Glanz werden von iOS automatisch hinzugefügt.¹⁰⁷

¹⁰⁴ Vgl. Windows Phone: UI Design and Interaction Guide for Windows Phone 7, S. 39

¹⁰⁵ Windows Phone: UI Design and Interaction Guide for Windows Phone 7, S. 38

¹⁰⁶ Vgl. Android Developer: Android Design, Iconography

¹⁰⁷ Vgl. Apple: iOS Human Interface Guidelines, Custom Icon and Image Creation Guidelines

Für Android muss das Symbol 48 x 48 Pixel¹⁰⁸ messen, für Windows Phone hingegen 173 x 173 Pixel. Bei Windows Phone ist es weiterhin möglich, Texte und Bilder im Symbol zu aktualisieren, um den Benutzer auf Änderungen oder neue Nachrichten hinzuweisen.¹⁰⁹



Abbildung 7 Startbildschirm von iOS, Android und Windows Phone 7 im Vergleich¹¹⁰

Navigation Bar

Unter iOS ist die Navigation Bar im oberen Bereich des Bildschirms zu finden und enthält immer den Titel der aktuellen Ansicht. Über den linken Button kann der Nutzer zurück zur letzten Ansicht navigieren. Dieser Button sollte daher auch stets mit dem Titel der vorherigen Ansicht beschriftet sein. Außerdem kann die Navigation Bar einen Button zum Bearbeiten von Inhalten enthalten.¹¹¹ In Abbildung 8 ist eine Ansicht des Kalenders unter iOS zu sehen. Mit Hilfe des Buttons mit dem Plus-Symbol können zum Beispiel neue Termine hinzugefügt werden.

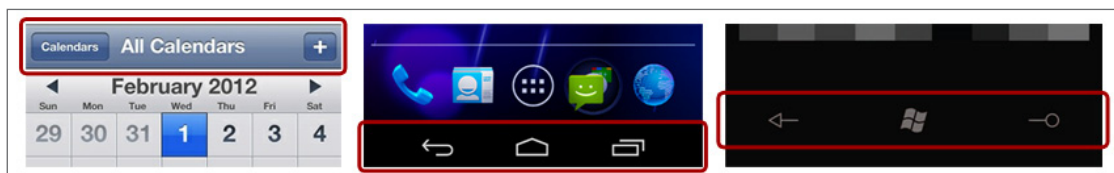


Abbildung 8 Navigation Bar unter iOS, Android und Windows Phone 7¹¹²

Unter Android und Windows Phone ist die Navigation Bar anwendungsübergreifend unten angeordnet, wie in Abbildung 8 zu sehen. Android unterscheidet zusätzlich zwischen Back- und Up-Button. Der Back-Button wird systemweit eingesetzt und bringt den Nutzer

¹⁰⁸ Vgl. Android Developer: Android Design, Iconography

¹⁰⁹ Vgl. Windows Phone: UI Design and Interaction Guide for Windows Phone 7, S. 47

¹¹⁰ Apple: iPhone, iOS

Android Developer: Android Design, UI Overview

Windows Phone: UI Design and Interaction Guide for Windows Phone 7, S. 26

¹¹¹ Vgl. Apple: iOS Human Interface Guidelines, User Experience Guidelines

¹¹² Apple: iOS Human Interface Guidelines, User Experience Guidelines

Android Developer: Android 4.0, Ice Cream Sandwich

Windows Phone: UI Design and Interaction Guide for Windows Phone 7, S. 16

schrittweise zurück. Der Up-Button ist in der Action Bar zu finden, die im nächsten Kapitel beschrieben wird. Dieser wird nur innerhalb der Anwendungen verwendet.¹¹³

Toolbar / Action Bar

Die Toolbar unter iOS ist beim iPod Touch und iPhone immer im unteren Bereich des Bildschirms angeordnet. Beim iPad ist auch eine Platzierung im oberen Bereich zusammen mit der Navigation Bar möglich. Die Toolbar enthält Aktionen die auf die aktuelle Ansicht bezogen sind.¹¹⁴ In Abbildung 9 links oben ist zum Beispiel die Toolbar der Mail-App zu sehen. Diese Funktionen stehen nur bei der Ansicht einer E-Mail zur Verfügung. Die Nachricht kann aktualisiert, verschoben und gelöscht werden. Weiterhin beinhaltet die Toolbar eine Funktion um auf die E-Mail zu antworten oder diese weiterzuleiten sowie eine neue Nachricht zu verfassen.

Unter Android sind die Funktionen, welche die aktuelle Ansicht betreffen, in der Action Bar platziert. Diese kann sowohl oben als auch unten angeordnet werden. Die Anordnung der Funktionen oben ist nur bei ausreichend Platz möglich, zum Beispiel im Landscape Modus oder auf Tablets. Die Action Bar im oberen Bereich enthält wenn verfügbar den Up-Button zum Navigieren in die übergeordnete Ansicht der App. Google stellt für die Funktionen innerhalb der Action Bar vordefinierte Icons zur Verfügung, die in den meisten Fällen genutzt werden sollten. Selten genutzte Funktionen sollten im Action Overflow positioniert sein.¹¹⁵ In Abbildung 9 ist dieser Bereich an den drei Punkten zu erkennen.

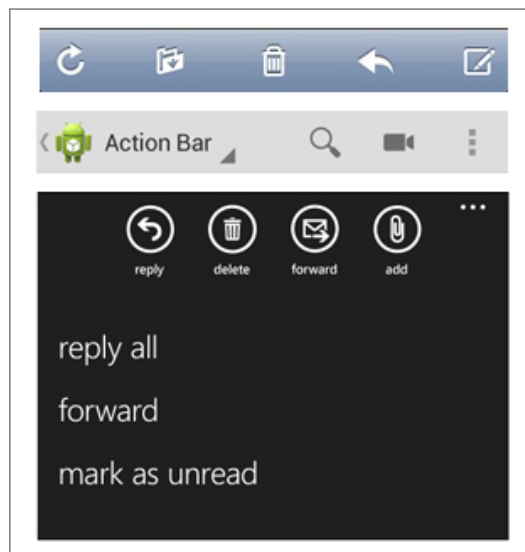


Abbildung 9 Toolbar unter iOS, Android und Windows Phone 7¹¹⁶

113 Vgl. Android Developer: Android Design, Navigation

114 Vgl. Apple: iOS Human Interface Guidelines, iOS UI Element Usage Guidelines

115 Vgl. Android Developer: Android Design, Action Bar

116 Apple: iOS Human Interface Guidelines, iOS UI Element Usage Guidelines

Android Developer: Android Design, Action Bar

Windows Phone: UI Design and Interaction Guide for Windows Phone 7, S. 36

Wichtige Funktionen werden unter Windows Phone in der Application Bar platziert. Hier können bis zu vier häufig genutzte Optionen untergebracht werden. Über die drei Punkte können bei Bedarf, wie auch in Abbildung 9 links unten zu sehen, weitere aufgerufen werden. Die Application Bar bleibt in jeder Orientierung stets an der Navigation Bar platziert. Die Icons passen sich der Position des Gerätes an.¹¹⁷

Tab Bar

Eine eigentliche Tab Bar gibt es grundsätzlich nur unter iOS. Dort ist diese immer unten angeordnet und bietet einen Wechsel zwischen verschiedenen Unteraufgaben, Ansichten oder Zuständen an. In der Uhr-App wechselt man zum Beispiel zwischen Weltuhr, Wecker, Stoppuhr und Timer mit Hilfe der Tab Bar. Dies ist auch in Abbildung 10 zu sehen. Wichtig ist vor allem das Hervorheben der aktuellen Position.¹¹⁸

Auf dem iPhone und iPod Touch können nicht mehr als fünf Tabs angezeigt werden. Sollte es mehrere Kategorien geben, können mit dem 5. Tab „Mehr“ weitere aufgerufen werden. Eine Sortierung durch den Benutzer ist möglich. Auf dem iPad können bis zu sieben Tabs angezeigt werden. Auf ein weiteres Aufrufen von Kategorien sollte verzichtet werden.¹¹⁹

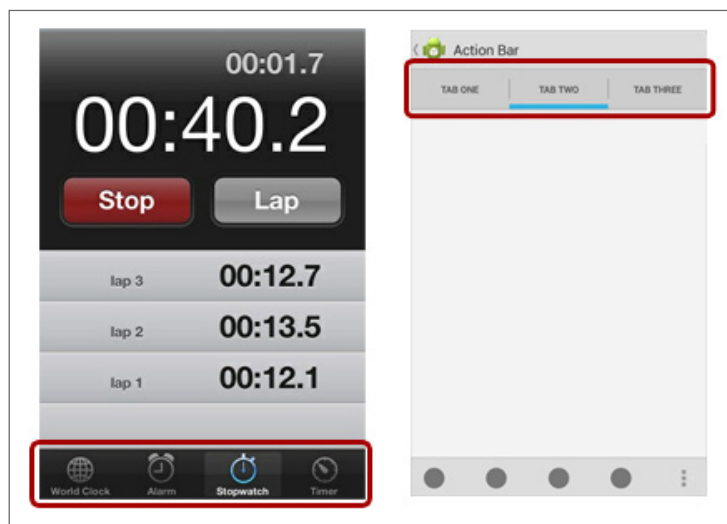


Abbildung 10 Tab Bar unter iOS und Android¹²⁰

Tabs unter Android befinden sich im Gegensatz zu iOS in der Action Bar und werden immer oben platziert, so wie im rechten Bereich von Abbildung 10 zu sehen. Neben den normalen, fixierten Tabs können weiterhin scrollbare Tabs genutzt werden. Scrollbare Tabs sollten verwendet werden, wenn viele Ansichten angezeigt werden müssen oder die Anzahl der Ansichten aufgrund des dynamischen Aufbaus nicht bekannt ist. Dabei ist zu beachten, dass

¹¹⁷ Vgl. Windows Phone: UI Design and Interaction Guide for Windows Phone 7, S. 31

¹¹⁸ Vgl. Apple: iOS Human Interface Guidelines, iOS UI Element Usage Guidelines

¹¹⁹ Vgl. Apple: iOS Human Interface Guidelines, iOS UI Element Usage Guidelines

¹²⁰ Apple: iOS Human Interface Guidelines, iOS UI Element Usage Guidelines
Android Developer: Android Design, Action Bar

nicht nur die Tab-Leiste an sich scrollbar ist, sondern dass der Benutzer auch durch Wischen nach Links oder Rechts im Inhaltsbereich zwischen den Ansichten wechseln kann.¹²¹

Unter Windows Phone gibt es eine extra Ansicht, die eine ähnliche Funktion wie die Tab Bar bietet. Dies ist die in Abbildung 11 dargestellte Pivot Control. Die Navigation funktioniert ähnlich der scrollbaren Tab Bar in Android. Der Nutzer wechselt zwischen den verschiedenen Ansichten durch Wischen nach Links oder Rechts im Inhaltsbereich.¹²²

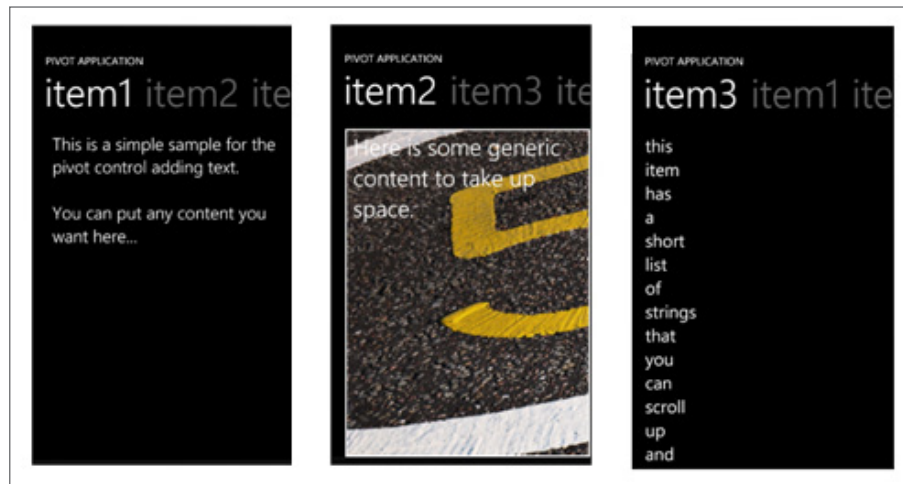


Abbildung 11 Pivot Control unter Windows Phone 7¹²³

Besonderheit Windows Phone: Panorama Control

Die Panorama Ansicht ist eine spezielle Möglichkeit zur Darstellung von Informationen, die nur für Windows Phone verfügbar ist. Anstatt Inhalte genau passend für den Bildschirm des Endgerätes zu gestalten, wird an dieser Stelle eine breite Leinwand angeboten, von der jeweils nur ein Ausschnitt zu sehen ist. Um dem Benutzer zu verdeutlichen, dass weitere Inhalte existieren, werden stets 24 Pixel des nächsten Abschnittes angezeigt.¹²⁴

In Abbildung 12 ist ein Beispiel für die Panorama Control mit Hintergrundbild zu sehen, einfarbige Hintergründe sind aber ebenfalls möglich. Bei der Verwendung eines Hintergrundbildes sollte stets auf die Lesbarkeit des Textes geachtet werden.

¹²¹ Vgl. Android Developer: Android Design, Action Bar

¹²² Vgl. Windows UI Design and Interaction Guide for Windows Phone 7, S. 175

¹²³ Microsoft: User Experience Design Guidelines, Pivot Control Design Guidelines for Windows Phone

¹²⁴ Vgl. Microsoft: User Experience Design Guidelines, Panorama Control Overview for Windows Phone

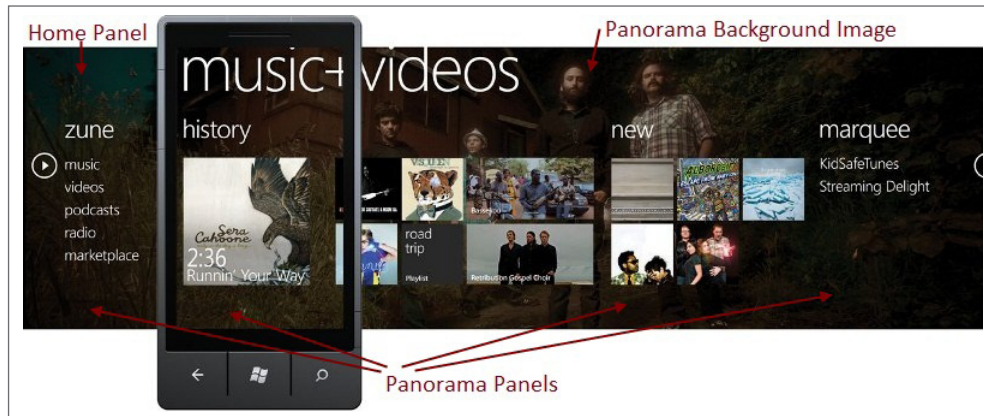
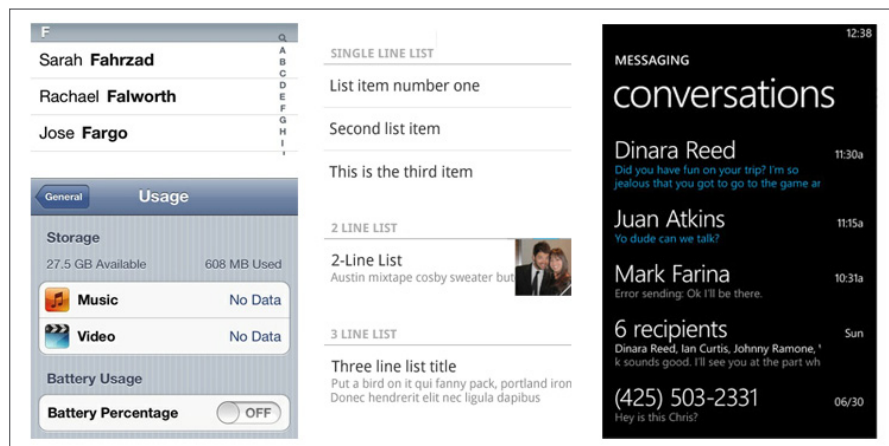
Abbildung 12 Panorama Control unter Windows Phone 7¹²⁵

Table View / List View

Um größere Mengen an Daten benutzerfreundlich darzustellen, sollten die Table View unter iOS bzw. die Listen unter Android und Windows genutzt werden. In Abbildung 13 sind die Elemente der jeweiligen Systeme zu sehen.

Unter iOS gibt es zahlreiche Darstellungsmöglichkeiten der Tabellen. Im linken Bereich in Abbildung 13 sind zum Beispiel eine einfache Liste mit einer alphabetischen Sortierung und eine in Sektionen eingeteilte Ansicht mit Überschrift und Beschreibung zu sehen. Neben benutzerdefinierten Bildern können auch Symbole wie zum Beispiel Häkchen, Pfeile oder Plus- und Minus-Icons zu den einzelnen Zeilen hinzugefügt werden. Wichtig ist das Anzeigen von Feedback bzw. das sofortige Umschalten der Ansicht, wenn der Nutzer eine Zeile selektiert hat.¹²⁶

Abbildung 13 Listen und Tabellen unter iOS, Android und Windows Phone 7¹²⁷

125 Microsoft: User Experience Design Guidelines, Central Application Hub with Home Page Menu (Panorama or Pivot Control) for Windows Phone

126 Vgl. Apple: iOS Human Interface Guidelines, iOS UI Element Usage Guidelines

127 Apple: iOS Human Interface Guidelines, iOS UI Element Usage Guidelines

Android Developer: Android Design, Lists

Microsoft: User Experience Design Guidelines, List with Details Drilldown for Windows Phone

Android verwendet Listen um Informationen zu selektieren sowie für aufklappbare Navigationen. Damit die Listen auch optisch in Sektionen eingeteilt sind, werden Trennlinien verwendet. Die einzelnen Zeilen können verschiedene Datentypen anzeigen. In Abbildung 13 sind Einträge mit Beschreibungen und Bildern zu sehen. Außerdem können Icons, Check-boxen und Buttons für Aktionen angezeigt werden.¹²⁸

Weiterhin können auch Grid Lists verwendet werden, wenn vordergründig Bilder angezeigt werden sollen.¹²⁹ In Abbildung 14 ist ein Ausschnitt einer solchen Liste in Form einer Bildergalerie zu sehen.

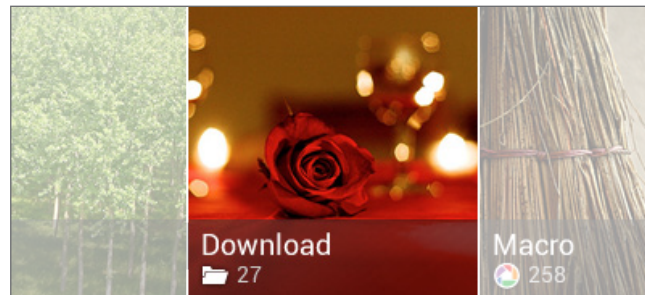


Abbildung 14 Grid List unter Android¹³⁰

Unter Windows Phone ist die Listen-Ansicht eine Übersicht von Daten, die der Nutzer selektieren kann um detaillierte Informationen anzuzeigen. Auch hier können Labels, Beschreibungen und Bilder genutzt werden.¹³¹

Auswahllisten

Auswahllisten in Form eines Popover gibt es in diesem Sinne nur für iOS auf dem iPad und unter Android. Mit Hilfe dieses Elements kann schnell eine Option aus einem Set ausgewählt werden.¹³²

Android bietet die Auswahllisten in Form von Drop-Down-Menüs an. Wenn das Menü geschlossen ist, wird immer der aktuell ausgewählte Eintrag angezeigt. Verwendet wird dieses Element in der Action Bar zum Umschalten von verschiedenen Ansichten oder in Formularen zum Auswählen aus vorgegebenen Einträgen.¹³³ In Abbildung 15 ist eine Auswahlliste unter Android im rechten Bereich zu sehen.

¹²⁸ Vgl. Android Developer: Android Design, Lists

¹²⁹ Vgl. Android Developer: Android Design, Grid Lists

¹³⁰ Android Developer: Android Design, Grid Lists

¹³¹ Vgl. Microsoft: User Experience Design Guidelines, List with Details Drilldown for Windows Phone

¹³² Vgl. Android Developer: Android Design, Spinners

¹³³ Vgl. Android Developer: Android Design, Spinners

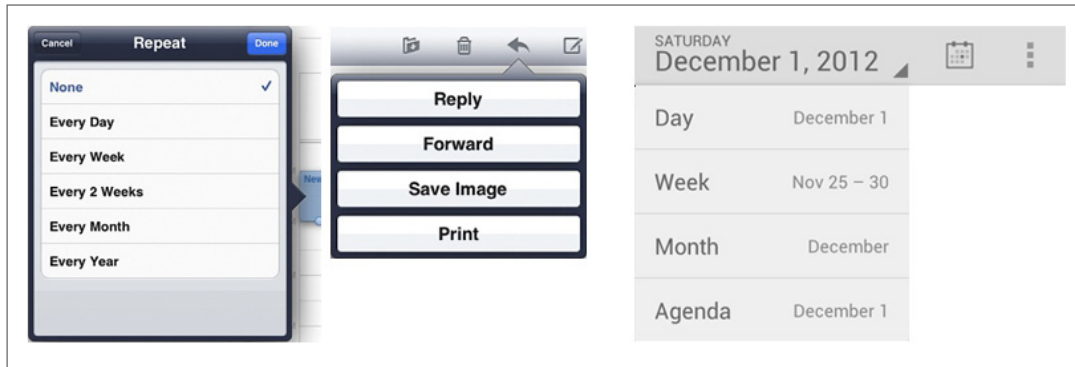


Abbildung 15 Actions Sheets unter iOS und Auswahlliste unter Android¹³⁴

Unter iOS können in diesem Popover noch weitere Elemente untergebracht sein, wie zum Beispiel Tabellen, Bilder, Text oder, wie in Abbildung 15 zu sehen, Buttons die kontextbezogen Funktionen auslösen. Diese Art des Overlays wird auch Action Sheet genannt.¹³⁵



Abbildung 16 Action Sheet auf dem iPhone¹³⁶

Beim iPhone und iPod Touch werden solche Action Sheets wie in Abbildung 16 am unteren Bildschirmrand untergebracht. Wählt der Benutzer eine Funktion aus, scrollt der Bereich von unten nach oben in das Sichtfeld. Der Hintergrund kann nicht bedient werden, bevor das Action Sheet geschlossen wird. Daher sollte der Hintergrund auch abgedunkelt dargestellt sein. Wichtig ist an dieser Stelle ein Abbrechen-Button, um wieder zurück zur vorherigen Ansicht zu kommen.¹³⁷

Action Sheets sollten stets ohne Scrollen bedienbar sein, die anzuzeigende Anzahl an Buttons ist also begrenzt. Um dem Benutzer deutlich zu zeigen, dass ein Button eine schwerwiegende Funktion hat, kann dieser rot eingefärbt werden.

134 Apple: iOS Human Interface Guidelines, iOS UI Element Usage Guidelines
Android Developer: Android Design, Spinners

135 Vgl. Apple: iOS Human Interface Guidelines, iOS UI Element Usage Guidelines

136 Apple: iOS Human Interface Guidelines, iOS UI Element Usage Guidelines

137 Vgl. Apple: iOS Human Interface Guidelines, iOS UI Element Usage Guidelines

Formulare / Texteingaben

Große Mengen an Eingaben sollten grundsätzlich vermieden werden, da die Nutzung der Tastatur durch die Größe der Endgeräte oft schwierig ist. Muss der Benutzer trotzdem Daten eingeben, sollte er optimal durch Vorgaben und Beschriftungen unterstützt werden.

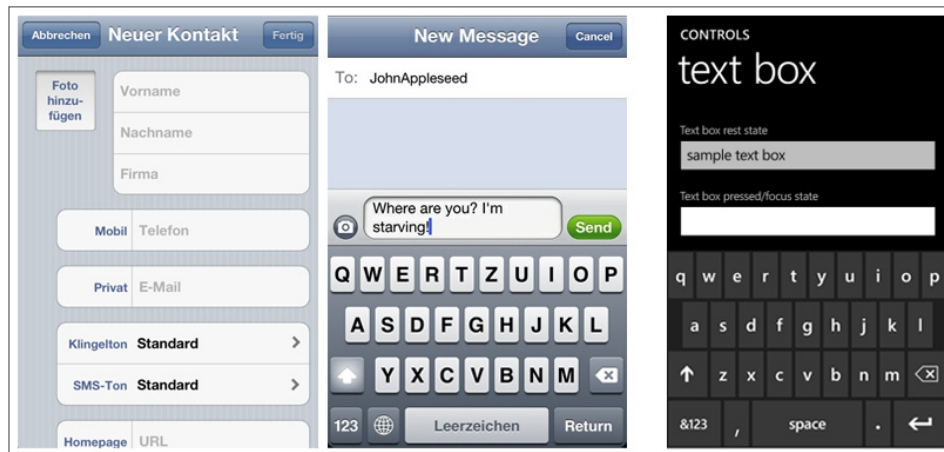


Abbildung 17 Formulare und Texteingaben unter iOS und Windows Phone 7¹³⁸

In Abbildung 17 und Abbildung 18 sind die Eingabefelder und Tastaturen der unterschiedlichen Betriebssysteme zu sehen. Diese unterscheiden sich größtenteils nur im Aussehen, die Funktion und die Handhabung sind ähnlich.

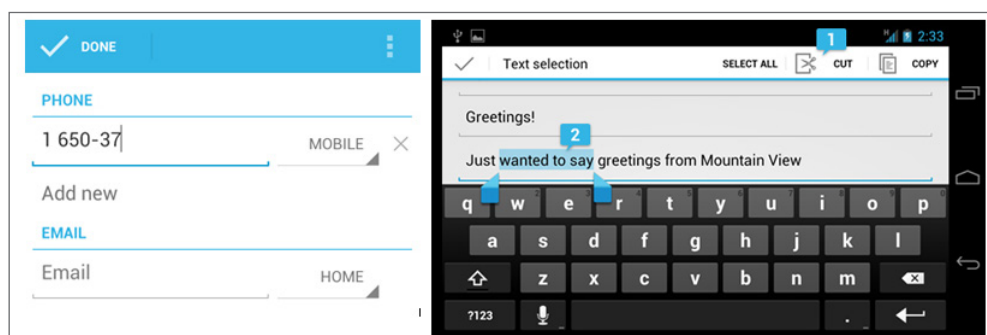


Abbildung 18 Formulare und Texteingaben unter Android¹³⁹

138 Apple: iOS Human Interface Guidelines, iOS UI Element Usage Guidelines
Microsoft: User Experience Design Guidelines, TextBox Control Design Guidelines for Windows Phone

139 Android Developer: Android Design, Text Fields

2.2 Accessibility - Richtlinien für die Zugänglichkeit

2.2.1 Überblick

Accessibility wird mit Barrierefreiheit oder Zugänglichkeit übersetzt. Der Anspruch besteht darin, dass alle Nutzer ohne fremde Hilfe und ohne Schwierigkeiten in allen Lebensbereichen handeln können. Der §4 des Behindertengleichstellungsgesetzes (BGG) drückt das wie folgt aus:

„Barrierefrei sind bauliche und sonstige Anlagen, Verkehrsmittel, technische Gebrauchsgegenstände, Systeme der Informationsverarbeitung, akustische und visuelle Informationsquellen und Kommunikationseinrichtungen sowie andere gestaltete Lebensbereiche, wenn sie für behinderte Menschen in der allgemein üblichen Weise, ohne besondere Erschwernis und grundsätzlich ohne fremde Hilfe zugänglich und nutzbar sind.“¹⁴⁰

Auf das Internet bezogen umfasst Accessibility die Nutzbarkeit von Webseiten und Webanwendungen durch Menschen mit Behinderungen. Diese müssen eine Webseite wahrnehmen und erkennen, sowie verstehen und mit dieser interagieren können.¹⁴¹ Auch wenn hier der Fokus auf Menschen mit Behinderungen liegt, profitieren grundsätzlich alle Nutzer von diesen Optimierungen. Besondere Vorteile haben außerdem ältere Benutzer sowie Benutzer mit eingeschränktem Lesevermögen oder Fremdsprachige. Auch Nutzer ohne Erfahrungen mit dem Internet profitieren von der Benutzerfreundlichkeit und Zugänglichkeit einer Webseite.¹⁴²

Das folgende Zitat hebt den Zusammenhang zwischen Usability und Accessibility hervor. Eine Webseite sollte benutzerfreundlich gestaltet und demzufolge effektiv, effizient und zufriedenstellend sein. Im Rahmen der Zugänglichkeit wird dieser Anspruch erweitert, sodass auch Benutzer mit Einschränkungen und Nutzer von assistiven Technologien die Verwendung einer Webseite als effektiv, effizient und zufriedenstellend erleben.

„Put more simply, usability means designing your website to be effective, efficient, and satisfying. Accessibility makes sure it is effective, efficient, and satisfying for more people, especially people with disabilities, in more situations - including with assistive technologies.“¹⁴³

2.2.2 Web Content Accessibility Guidelines (WCAG) 2.0

Bei der WCAG 2.0 handelt es sich um eine Sammlung von Richtlinien und Empfehlungen, um Inhalte im Internet zugänglich zu gestalten. Das Dokument wurde vom World Wide Web Consortium verfasst und ist seit dem 11. Dezember 2008 ein anerkannter Standard.¹⁴⁴

¹⁴⁰ Bundesministerium der Justiz: Gesetz zur Gleichstellung behinderter Menschen, § 4 Barrierefreiheit

¹⁴¹ Vgl. Thatcher ; Burks ; Heilmann 2006, S. 2

¹⁴² Vgl. Thatcher ; Burks ; Heilmann 2006, S. 3, 8

¹⁴³ Thatcher ; Burks ; Heilmann 2006, S. 27

¹⁴⁴ Vgl. W3C: Web Content Accessibility Guidelines (WCAG) 2.0

Die einzelnen Kriterien innerhalb der WCAG 2.0 werden anhand von unterschiedlichen Punkten in drei Stufen eingeteilt. Stufe A ist dabei die niedrigste, Stufe AAA die höchste. Zunächst muss definiert werden, ob ein Kriterium zwingend notwendig für die Zugänglichkeit einer Webseite ist. Weiterhin fließt die Erfüllbarkeit dieses mit in die Bewertung ein. Ist ein Kriterium grundsätzlich für alle Arten von Webseiten erfüllbar, bekommt es eine niedrigere Stufe als ein Kriterium, welches nur mit Hilfe spezieller Technologien umgesetzt werden kann.¹⁴⁵

Die WCAG 2.0 setzt sich aus den vier Richtlinien zusammen, die im Folgenden erläutert werden.

Prinzip 1 - Wahrnehmbar	
1.1 Text-Alternativen	Stellen Sie Text-Alternativen für alle Nicht-Text-Inhalte zur Verfügung, so dass diese in andere vom Benutzer benötigte Formen geändert werden können, wie zum Beispiel Großschrift, Braille, Symbole oder einfachere Sprache.
1.2 Zeitbasierte Medien	Stellen Sie Alternativen für zeitbasierte Medien zur Verfügung
1.3 Anpassbar	Erstellen Sie Inhalte, die auf verschiedene Arten dargestellt werden können (z.B. einfacheres Layout), ohne dass Informationen oder Struktur verloren gehen.
1.4 Unterscheidbar	Machen Sie es Benutzern leichter, Inhalt zu sehen und zu hören einschließlich der Trennung von Vorder- und Hintergrund.

Tabelle 5 WCAG 2.0: Prinzip 1 - Wahrnehmbar¹⁴⁶

Das erste Prinzip innerhalb der WCAG 2.0 bezieht sich auf die Wahrnehmbarkeit von Inhalten. In Tabelle 5 sind die vier Kriterien zusammengefasst.

Im Rahmen der Zugänglichkeit geht es vor allem darum, alternative Inhalte anzubieten. Diese können von assistiven Technologien wie zum Beispiel Screenreadern für den Benutzer ausgegeben werden. Dabei sind alternative Inhalte für alle Arten von Medien, wie zum Beispiel für Bilder oder Videos zu hinterlegen.

Weiterhin müssen dem Benutzer Zusammenhänge verdeutlicht werden, die dieser wegen seiner Behinderung nicht wahrnehmen kann. Zum Beispiel ist die Verbindung von Formularelementen und deren Beschriftung sowie zugehörigen Hinweisen nicht einfach durch die visuelle Gestaltung klar. Assistive Technologien müssen dies erkennen und an den Benutzer weiter geben können. Um dies zu realisieren sind unter anderem semantische Elemente, wie Überschriften oder Labels, sowie grundsätzlich valider Code zu verwenden.

¹⁴⁵ Vgl. W3C: Understanding WCAG 2.0, A guide to understanding and implementing Web Content Accessibility Guidelines 2.0

¹⁴⁶ W3C: Web Content Accessibility Guidelines (WCAG) 2.0, Autorisierte deutsche Übersetzung

Prinzip 2 - Bedienbar	
2.1 Per Tastatur zugänglich	Sorgen Sie dafür, dass alle Funktionalitäten per Tastatur zugänglich sind.
2.2 Ausreichend Zeit	Geben Sie den Benutzern ausreichend Zeit, Inhalte zu lesen und zu benutzen.
2.3 Anfälle	Gestalten Sie Inhalte nicht auf Arten, von denen bekannt ist, dass sie zu Anfällen führen.
2.4 Navigierbar	Stellen Sie Mittel zur Verfügung, um Benutzer dabei zu unterstützen zu navigieren, Inhalte zu finden und zu bestimmen, wo sie sich befinden.

Tabelle 6 WCAG 2.0: Prinzip 2 - Bedienbar¹⁴⁷

Das zweite Prinzip der WCAG 2.0 ist in Tabelle 6 dargestellt und beschreibt die Bedienbarkeit von Webseiten. Grundsätzlich müssen alle Benutzer eine Webseite unabhängig von Ein- oder Ausgabegeräten bedienen können. Nicht alle Benutzer können zum Beispiel eine Maus verwenden oder die Ausgaben auf einem Bildschirm sehen. Besonders die Tastatur als alternatives Eingabemittel wird durch die WCAG 2.0 hervorgehoben.

Im Rahmen der Gebrauchstauglichkeit wurde bereits die Wichtigkeit einer richtigen Gestaltung der Navigation erläutert. Für die Barrierefreiheit sind weitere Optimierungen notwendig, wie zum Beispiel die Möglichkeit bestimmte Elemente oder Gruppen zu überspringen. Grundsätzlich muss sich der Benutzer jederzeit orientieren können und den Überblick über die Navigationsoptionen behalten.

Prinzip 3 - Verständlich	
3.1 Lesbar	Machen Sie Inhalt lesbar und verständlich.
3.2 Vorhersehbar	Sorgen Sie dafür, dass Webseiten vorhersehbar aussehen und funktionieren.
3.3 Hilfestellung bei der Eingabe	Helfen Sie den Benutzern dabei, Fehler zu vermeiden und zu korrigieren.

Tabelle 7 WCAG 2.0: Prinzip 3 - Verständlich¹⁴⁸

Bei dem Kriterium Verständlichkeit, welches in Tabelle 7 dargestellt ist, geht es vordergründig um die Lesbarkeit von Inhalten. Zunächst sollte immer die Muttersprache des Benutzers zur Verfügung stehen. Die Sprache muss zwingend im HTML ausgezeichnet werden, damit assistive Technologien diese erkennen und die Texte entsprechend vorlesen können. Abbildung 19 zeigt die Auszeichnung über das Attribut `lang`.

147 W3C: Web Content Accessibility Guidelines (WCAG) 2.0, Autorisierte deutsche Übersetzung

148 W3C: Web Content Accessibility Guidelines (WCAG) 2.0, Autorisierte deutsche Übersetzung

```
<html lang="de">  
<!-- Inhalt -->  
</html>
```

Abbildung 19 Auszeichnung der Sprache über das Attribut lang

Informationen sollten stets in einfacher Sprache verfasst sein. Dafür ist es notwendig auf Fremdwörter oder eine technische Sprache zu verzichten. Das W3C schlägt an dieser Stelle die Nutzung des Leseniveaus der niedrigen sekundären Schulbildung vor. Für Benutzer mit Leseschwäche oder einer Lernbehinderung sowie für fremdsprachige Benutzer sind diese Texte dann leichter zu verstehen.

Die Richtlinien 3.2 und 3.3 beziehen sich stark auf die Kriterien Erwartungskonformität und Fehlertoleranz, die auch für die Zugänglichkeit von großer Bedeutung sind.

Prinzip 4 - Robust	
4.1 Kompatibel	Maximieren Sie die Kompatibilität mit aktuellen und zukünftigen Benutzer-Agenten, einschließlich assistierender Techniken.

Tabelle 8 WCAG 2.0: Prinzip 4 - Robust¹⁴⁹

In Tabelle 8 ist die Anforderung an Robustheit von Webseiten dargestellt. Diese sollten mit möglichst vielen Benutzeragenten kompatibel sein. Dies umfasst nicht nur die verbreiteten Browser sondern auch assistive Technologien wie Screenreader oder Braillezeilen sowie mobile Endgeräte.

2.2.3 Zugänglichkeit von Geräten mit Touchscreen

Besonders für Nutzer mit Einschränkungen des Sehsinns und mit motorischen Problemen, aber auch mit einer Behinderung des Gehörs ist die Bedienung eines Touchscreens zunächst schwierig. Ausgewählte Technologien ermöglichen es diesen Menschen aber trotzdem, mobile Geräte effektiv zu verwenden. Vor allem Apple und Google bieten im Rahmen ihrer Betriebssysteme eine große Auswahl an Accessibility-Funktionen an. Diese Technologien werden in Kapitel 3.4 näher besprochen.

Problematisch bei der Nutzung eines Touchscreens ist zunächst die fehlende physische Tastatur, mit Hilfe derer die Benutzer die Tasten erfühlen können. Durch Gewohnheit und Übung kennen die Benutzer meist die Position der einzelnen Tasten und können auch ohne die Fähigkeit des Sehens schreiben. Neben herkömmlichen Tastaturen besteht auch die Möglichkeit, eine Tastatur mit Blindenschrift zu verwenden. Ein weiterer Vorteil einer physischen Tastatur ist die haptische Rückmeldung. Der Benutzer hört und fühlt etwas beim Betätigen einer Taste. Bei der Tastatur eines Touchscreens können zwar Töne als Feedback abgespielt werden, aber ein Reproduzieren der Haptik oder die Bereitstellung einer Tastatur mit Blindenschrift ist nicht möglich. Zwar besteht für die Benutzer die Möglichkeit, physische

149 W3C: Web Content Accessibility Guidelines (WCAG) 2.0, Autorisierte deutsche Übersetzung

Tastaturen an Geräte mit Touchscreen anzuschließen, dadurch geht allerdings ein Teil der Mobilität verloren.

Ein weiteres Problem bei der Eingabe von Text ist die geringe Größe der Tasten. Der Benutzer muss seine Finger relativ präzise positionieren, um den gewünschten Buchstaben zu treffen. Selbst Benutzern ohne Einschränkungen gelingt diese Präzision nicht in jedem Fall. Sonstige Elemente mit denen der Benutzer interagiert, können größer gestaltet werden um die Bedienung mit den Fingern zu erleichtern. Benutzern mit Einschränkungen der motorischen Fähigkeiten oder des Sehsinns wird es trotzdem schwer fallen diese Elemente zu betätigen. Hier lassen sich Parallelen zur Verwendung einer Maus am Computer feststellen. In diesem Fall verwenden die betreffenden Nutzer die Tastatur zur Navigation innerhalb des Systems. Da die Tastatur bei einem Touchscreen aber ebenfalls nicht ideal ist, müssen Alternativen gefunden werden.

Eine sinnvolle Alternative ist die Bereitstellung eines Screenreaders. Apple und Google bieten in ihren Systemen iOS und Android standardmäßig einen solchen an. Beispielhaft soll im Folgenden die grundlegende Nutzung von VoiceOver, dem Screenreader unter iOS, beschrieben werden.

Grundsätzlich liest ein Screenreader die Inhalte und Elemente auf dem Bildschirm vor. VoiceOver ermöglicht zusätzlich für den Benutzer eine andere Art der Navigation. Um das mobile Gerät zu kontrollieren werden einfache und vom sonstigen Gebrauch abweichende Gesten genutzt. Zum Beispiel muss der Benutzer einmal den Bildschirm berühren, damit Informationen des ausgewählten Objektes vorgelesen werden. Um dieses Element auch zu aktivieren, muss ein doppeltes Antippen erfolgen.¹⁵⁰ Neben einem freien Auswählen von Elementen auf dem Bildschirm kann auch mittels Wischen zum nächsten oder vorherigen Element navigiert werden. Auf diese Weise ist es für den Benutzer möglich, auch kleinere, schwer auszuwählende Komponenten zu aktivieren.¹⁵¹

Damit die Screenreader unter iOS und Android zuverlässig und für den Benutzer zufriedenstellend arbeiten können, müssen bestimmte Besonderheiten bei der Erstellung von Webseiten und nativen Apps beachtet werden. Diese werden in Kapitel 3.5 besprochen.

150 Vgl. Apple: Accessibility, iPhone

151 Vgl. Apple: Accessibility, iPhone

3 Moderne Webtechnologien

Bei der Bereitstellung von Inhalten auf mobilen Endgeräten muss zunächst ausgewählt werden, ob eine mobile Webseite für den jeweiligen Zweck und für die Benutzer ausreichend ist oder ob eine native Applikation (App) zum Einsatz kommen soll. Die mobile Webseite kann grundsätzlich von allen Geräten über einen Browser aufgerufen werden. Die native App muss für jedes Betriebssystem, das man unterstützen möchte, eigens erstellt werden. Im folgenden Kapitel werden die Vor- und Nachteile der beiden Technologien diskutiert. Anschließend erfolgt eine Vorstellung der Technologien zur Erstellung mobiler Webseiten sowie nativer Apps.

Weiterhin wird HTML5, die neue Version der Auszeichnungssprache HTML, sowie deren Vorteile vorgestellt. HTML5 kann sowohl im Desktop- als auch im mobilen Kontext verwendet werden.

3.1 HTML5

Die Auszeichnungssprache HTML5 ist eine Spezifikation der WHATWG¹⁵² und des W3C. Aktuell (Stand 2012) ist HTML5 kein Standard und befindet sich noch in der Bearbeitung.

Obwohl es sich bei HTML5 um einen Entwurf handelt, können zahlreiche Funktionen und Elemente bereits verwendet werden. HTML5 ist größtenteils abwärtskompatibel, d.h. Benutzer mit modernen Browsern profitieren bereits von den Vorteilen, Nutzer von älteren Technologien können Webseiten trotzdem ohne Einschränkungen verwenden.

In den folgenden Kapiteln werden ein Überblick über die Syntax gegeben sowie die neuen Sectioning Elemente und ausgewählte Schnittstellen vorgestellt.

3.1.1 Die HTML5-Syntax

Die Syntax von HTML5 ist vollständig kompatibel zu den Vorgängerversionen HTML 4.01 und XHTML 1. Wenn in einem HTML-Dokument der HTML5 Doctype für die Definition der Version und des Typs genutzt wird, liegt bereits valides HTML5 vor. Dieser Doctype funktioniert ohne Einschränkung in allen Browsern.¹⁵³ Eine Gegenüberstellung der Doctype-Definitionen von HTML 4.01 und HTML5 ist in Abbildung 20 zu sehen. Der Doctype für HTML5 wurde stark reduziert und ist somit einfacher zu verwenden.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
„http://www.w3.org/TR/html4/loose.dtd">

<!DOCTYPE html>
```

Abbildung 20 Doctype-Definition: HTML 4.01 und HTML5 im Vergleich

¹⁵² WHATWG: Web Hypertext Application Technology Working Group

¹⁵³ Vgl. Kröner 2011, S. 41, 44, 56

Neben dem Reduzieren des Doctypes können zum Beispiel auch die Elemente `<html>` und `<body>` weg gelassen werden. Weiterhin wird für eine Vielzahl von Elementen nur das Start-Tag benötigt. Das abschließende Tag kann somit entfallen. In Abbildung 21 ist ein formal korrektes HTML5-Dokument dargestellt, welches weitestgehend vereinfacht ist. Grundsätzlich ist die HTML-Syntax gleich geblieben und die reduzierte Variante muss nicht verwendet werden.

```
<!DOCTYPE html>
<head>
  <meta charset="utf-8">
  <title>Mein erstes HTML5-Dokument</title>
</head>
<h1>HTML5</h1>
<p>Hallo!
```

Abbildung 21 HTML5: Reduziertes Dokument¹⁵⁴

3.1.2 Sectioning Elemente

Bisher verwendete man für jeden notwendigen Inhaltsabschnitt ein `div`-Element, meist mehrfach verschachtelt. Durch diese Vorgehensweise wird die Anprechbarkeit über CSS und Javascript realisiert. Das `div`-Element hat allerdings keinerlei semantische Bedeutung und sollte laut W3C nur verwendet werden, wenn kein anderes Element für den jeweiligen Zweck passend ist.¹⁵⁵

Mit Hilfe der neuen HTML5 Sectioning Elemente entsteht eine eindeutige Struktur sowie semantisches Markup. Besonders Benutzer von assistiven Technologien profitieren von Elementen wie `<nav>`, `<section>` und `<header>`. Ein weiterer Vorteil ist die bessere Verarbeitung der Webseite durch Suchmaschinen.¹⁵⁶ In Tabelle 9 ist eine Auswahl der Sectioning Elemente aufgeführt und kurz beschrieben.

Element	Verwendung
<code><header></code>	Kopfbereich für die gesamte Seite oder für andere Elemente, Sammlung einführender Inhalte (Überschriften, Navigation)
<code><footer></code>	Fußbereich für die gesamte Seite oder für andere Elemente, Sammlung von Informationen wie Autor, Links, Copyright
<code><nav></code>	Hauptnavigation der Webseite, Liste von Links
<code><section></code>	Inhaltliche Gliederung, sollte mit einer Überschrift versehen werden
<code><article></code>	Auszeichnung in sich geschlossener Inhalte, die für sich alleine stehen können

Tabelle 9 HTML5: Übersicht Sectioning Elemente¹⁵⁷

¹⁵⁴ Kröner 2011, S. 56

¹⁵⁵ Vgl. W3C: HTML5, A vocabulary and associated APIs for HTML and XHTML, 26.6.2012

¹⁵⁶ Vgl. Kröner 2011, S. 39

¹⁵⁷ Vgl. Hogan 2011, S. 25ff; Lawson ; Sharp 2012, S. 13ff; Kröner 2011, S. 115ff

Die neuen Elemente werden bereits von allen modernen Browsern unterstützt. Ältere Versionen von unter anderem Firefox, Google Chrome und Internet Explorer kennen diese Elemente allerdings nicht. Um negative Effekte beim Layout zu vermeiden, sollte man die Anweisung aus Abbildung 22 verwenden, um die Elemente als Block-Elemente zu definieren. Damit verhalten sich die modernen und auch die älteren Versionen der Browser gleich.

```
header, footer, nav, section, article {  
  display: block;  
}
```

Abbildung 22 HTML5: CSS-Anweisung für die Unterstützung der Sectioning Elemente

Für den Internet Explorer bis Version 8 müssen die Elemente zusätzlich über Javascript definiert werden. Ansonsten kann dieser Browser die Anweisung aus Abbildung 22 nicht korrekt umsetzen. Somit kann es unter Umständen Schwierigkeiten mit dem Layout der Webseite geben. Die notwendige Javascript-Anweisung ist in Abbildung 23 zu sehen. Die if-Anweisung in der ersten Zeile schränkt die Verwendung der Anweisung auf die Versionen vor Internet Explorer 9 ein. Ab Version 9 werden die Sectioning Elemente erkannt und die CSS-Anweisung aus Abbildung 22 reicht für die korrekte Darstellung aus.

```
<!--[if lt IE 9]>  
<script>  
  document.createElement('header');  
  document.createElement('footer');  
  document.createElement('nav');  
  ...  
</script>  
<![endif]>
```

Abbildung 23 HTML5: Javascript-Anweisung für den Internet Explorer¹⁵⁸

3.1.3 Offline Web Applications

Für die Interaktion mit Webseiten und Web-Apps über den Browser auf dem mobilen Endgerät ist grundsätzlich eine Verbindung zum Internet notwendig. Bei der Nutzung eines Desktop-PCs oder des mobilen Gerätes zu Hause ist die Verbindung zum Internet meist vorhanden und auch stabil. Ist der Benutzer aber unterwegs kann dies nicht mehr garantiert werden, sodass die jeweilige Webseite unter Umständen nicht verwendbar ist.

Mit Hilfe der HTML5 Offline Web Applications können Teile oder die gesamte Webseite im Browser gespeichert werden, sodass diese auch offline zugänglich sind.¹⁵⁹ Das Speichern von unter anderem HTML-, CSS- und Javascript-Dateien ist möglich. Außerdem können auch Aktionen des Benutzers, die eigentlich einen Zugang zum Internet erfordern gespeichert werden. So kann der Benutzer zum Beispiel ein Formular ausfüllen und abschicken.

¹⁵⁸ Vgl. Lawson / Sharp 2012, S. 11 und Hogan 2011, S. 38

¹⁵⁹ Vgl. Kröner 2011, S. 225f

Das Formular wird zunächst im Browser gesichert und versendet, sobald wieder eine Verbindung zum Internet besteht.¹⁶⁰

Damit eine Webseite oder Webanwendung offline genutzt werden kann, muss dem Browser zunächst mitgeteilt werden, welche Dateien er speichern soll. Zu diesem Zweck ist eine Manifest-Datei zu erstellen. Dabei handelt es sich um eine Liste der Dateien, die gespeichert werden sollen und somit offline verfügbar sind.¹⁶¹ In Abbildung 24 ist beispielhaft die Manifest-Datei `clock.appcache` dargestellt.

```
CACHE MANIFEST
clock.html
clock.css
clock.js
```

Abbildung 24 HTML5: Offline Web Applications - Manifest¹⁶²

In der HTML-Datei muss zusätzlich eine kleine Anpassung erfolgen. Die Datei `clock.appcache` wird dem HTML-Element wie in Abbildung 25 zu sehen zugewiesen.

```
<!DOCTYPE HTML>
<html manifest="clock.appcache">
  <head>
    <title>Clock</title>
    <script src="clock.js"></script>
    <link rel="stylesheet" href="clock.css">
  </head>
  <body>
```

Abbildung 25 HTML5: Offline Web Applications - HTML-Datei¹⁶³

Zusätzlich zum Festlegen der Dateien, die gespeichert werden sollen, kann man dem Browser weiterhin mitteilen, welche Dateien er nicht offline verfügbar machen soll. Dies ist nützlich um sicherzustellen, dass bestimmte Teile der Seite nur zugänglich sind, wenn eine Internetverbindung besteht. Weiterhin können Dateien bestimmt werden, die anstelle dieser Inhalte bereitzustellen sind. Dies ist zum Beispiel bei einem Chat sinnvoll. Anstelle des Chats kann im Falle einer fehlenden Internetverbindung eine Informationsseite geladen werden, die den Benutzer über die Verfügbarkeit der Funktion aufklärt.¹⁶⁴

In Abbildung 26 ist die Ersetzung von Dateien zu sehen. Dieser sogenannte Fallback wird direkt mit in der Manifest-Datei notiert. In der zweiten Zeile wird die Datei `example.js` mit der Datei `fallback-example.js` ersetzt, wenn der Benutzer offline ist. Die Anweisung `„/ offline.html“` bestimmt diese HTML-Datei als Fallback für alle anderen Dateien. Die Datei `offline.html` wird demnach immer angezeigt, wenn eine Seite nicht erreicht werden kann, weil der Benutzer offline ist oder die entsprechende Datei nicht im Speicher liegt.¹⁶⁵

¹⁶⁰ Vgl. Kröner 2011, S. 226

¹⁶¹ Vgl. W3C: HTML5, A vocabulary and associated APIs for HTML and XHTML, 26.6.2012
Vgl. Lawson ; Sharp 2012, S. 208

¹⁶² W3C: HTML5, A vocabulary and associated APIs for HTML and XHTML, 26.6.2012

¹⁶³ W3C: HTML5, A vocabulary and associated APIs for HTML and XHTML, 26.6.2012

¹⁶⁴ Vgl. Lawson ; Sharp 2012, S. 208

¹⁶⁵ Vgl. Lawson ; Sharp 2012, S. 211


```
FALLBACK:
example.js fallback-example.js
/ offline.html

NETWORK:
*
```

Abbildung 26 HTML5: Offline Web Applications - Fallback und Network¹⁶⁶

Eine weitere Angabe in der Manifest-Datei ist Network, welche in Abbildung 26 zu sehen ist. Mit Hilfe dieser Angabe werden die Dateien aufgeführt, die nur online zugänglich sein sollen. Wird zum Beispiel der Stern (*) verwendet, sollen alle Dateien, die nicht unter Cache Manifest aufgeführt sind, nur bei vorhandener Internetverbindung verfügbar sein.¹⁶⁷

Die Funktion Offline Web Applications hat aktuell (Stand 2012) den Status Working Draft und befindet sich demnach noch im Entwurfsstadium. Die Spezifikation ist folglich weiteren Änderungen unterworfen. Trotzdem wird diese von Safari unter iOS und dem Android Browser sowie den Desktop Browsern Firefox, Chrome, Opera und Safari bereits unterstützt. Opera Mobile und der Internet Explorer bis Version 9 bieten noch keine Unterstützung an.¹⁶⁸

3.1.4 Geolocation API

Bei der Geolocation API handelt es sich um eine Schnittstelle, die das Erkennen der aktuellen Position eines mobilen Gerätes und damit des Benutzers ermöglicht. Google Maps nutzt diese Schnittstelle zum Beispiel bereits zur Lokalisierung der Benutzer.¹⁶⁹ Die Funktion umfasst nicht nur das einmalige Lokalisieren, sondern auch das Verfolgen der Position. Zu diesem Zweck wird der Standort in regelmäßigen Intervallen abgefragt.¹⁷⁰

Die Geolocation API wird in einer eigenen Spezifikation behandelt und hat aktuell den Status Proposed Recommendation. Dabei handelt es sich um einen Vorschlag zur Empfehlung und die endgültige Abnahme wird erwartet.¹⁷¹ Die Unterstützung der Browser ist ebenfalls weit fortgeschritten. Firefox, Chrome, Safari, Opera und der Internet Explorer ab Version 9 sowie die mobilen Browser unter iOS und Android unterstützen die Geolocation API bereits.¹⁷²

Zu beachten ist, dass der Benutzer immer darauf aufmerksam gemacht wird, dass eine Anwendung seine aktuelle Position nutzen möchte. Der Benutzer kann diese Anfrage auch ablehnen. Eine Anwendung sollte sich demnach nicht auf die Lokalisierung verlassen, sondern auch ohne diese Daten weiterhin funktionieren. Ist die Angabe der aktuellen Position wichtig, muss dem Benutzer eine alternative Eingabemöglichkeit für die Information angeboten werden.

¹⁶⁶ Lawson ; Sharp 2012, S. 210f

¹⁶⁷ Vgl. Lawson ; Sharp 2012, S. 212

¹⁶⁸ Vgl. Deveria: When can i use, Offline Web Applications

¹⁶⁹ Vgl. Lawson ; Sharp 2012, S. 238

¹⁷⁰ Vgl. Lawson ; Sharp 2012, S. 239

¹⁷¹ Vgl. W3C: Geolocation API Specification, 10.5.2012

¹⁷² Vgl. Lawson ; Sharp 2012, S. 238

3.2 Mobile Webseite oder native App

Webseiten oder Webanwendungen sind grundsätzlich auf allen Geräten verfügbar, da sie über den Browser aufgerufen werden können. Für mobile Geräte müssen zwar einige Anpassungen vorgenommen werden, der Aufwand ist an dieser Stelle aber vergleichsweise gering. Das Web ist weit verbreitet, standardisiert und hat die höchste Verfügbarkeit.¹⁷³ Webseiten sind relativ günstig zu erstellen sowie sehr einfach über das Web bereitzustellen.

Demgegenüber steht die Entwicklung von nativen Apps. Für alle Betriebssysteme, die unterstützt werden sollen, müssen völlig eigenständige Anwendungen erstellt werden. Die Palette an Systemen ist vielfältig und geht weit über iOS und Android hinaus. Weiterhin existieren viele verschiedene Versionen der einzelnen Systeme auf dem Markt und in Zukunft werden weitere hinzu kommen. Dies ist ein schwerwiegender Nachteil, da in den meisten Unternehmen nicht genügend Ressourcen für die Erstellung von mehreren nativen Apps zur Verfügung stehen. Vom erhöhten finanziellen Aufwand abgesehen, ist auch zusätzliches Wissen im Rahmen der Programmierung notwendig. Für iOS muss in Objective C, für Android in Java und für Windows Phone in C# entwickelt werden.

Weiterhin ist für den Vertrieb der Apps ein Einstellen in den jeweiligen Store der Hersteller Voraussetzung. Dies ist mit zusätzlichem Aufwand verbunden, da die Auflagen und Kontrollen der Anwendungen sehr hoch sind. Weiterhin müssen bestimmte Regeln und Rahmenbedingungen eingehalten sowie ein Teil der Einnahmen durch den Verkauf von Apps an den Betreiber des Stores abgegeben werden.¹⁷⁴ Durch dieses System entsteht grundsätzlich eine Abhängigkeit von den Herstellern der Betriebssysteme bzw. den Betreibern der Stores. Solche Abhängigkeiten gibt es im Gegensatz dazu bei Webseiten nicht.

Für den Benutzer haben native Apps einige Vorteile, die in die Auswahl der Technologie mit einbezogen werden sollten. Zum einen ist die Nutzung intuitiver, da Elemente auf gleiche Art und Weise gestaltet sind und funktionieren, wie der Rest des jeweiligen Systems. Dadurch können die Benutzer bereits erworbene Kenntnisse über Aussehen und Funktionen anwenden.¹⁷⁵

Außerdem kann mit einer nativen App auf alle Funktionen der Geräte zugegriffen werden, zum Beispiel auf die Kamera, den Lokalisierungsdienst sowie den eingebauten Bewegungsmesser.¹⁷⁶ Die Nutzung des Dateisystems ist bislang (Stand 2012) nur über eine native App möglich. Der Nutzer kann über eine App auf seine Fotos, das Adressbuch und auf die E-Mail Funktion zugreifen.¹⁷⁷ Da diese Daten privat und sehr sensibel sind, wäre eine Verwendung dieser auch über eine mobile Webseite kritisch zu betrachten.

173 Vgl. Fling 2009, S. 145

174 Vgl. Fling 2009, S. 146

175 Vgl. Apple: iOS Human Interface Guidelines, Human Interface Principles
Vgl. Hoover ; Berkman: Designing Mobile Interfaces, Principles of Mobile Design

176 Vgl. Fling 2009, S. 147f

177 Vgl. Fling 2009, S. 148f

Dagegen ist es bereits zum Teil möglich, die Kamera und den Lokalisierungsdienst auch über Webseiten zu benutzen. Das W3C entwickelt zur Zeit die Geolocation API¹⁷⁸, sowie eine Schnittstelle zum Zugriff auf die Kamera auch über Webseiten.¹⁷⁹

Neben der Möglichkeit, diese beiden Technologien getrennt zu betrachten, ist auch eine Kombination möglich. Bei dieser handelt es sich um Web Apps, die über den Browser aufgerufen werden, aber mehr wie eine native Anwendung aussehen und sich auch eher wie eine solche verhalten.¹⁸⁰ Auf diese Weise können die Vorteile von beiden Technologien vereint werden.

Problematisch kann dies allerdings für den Benutzer sein, der durch Aussehen und Verhalten eigentlich zwischen Webseite und nativer Anwendung unterscheidet. Durch die Verwendung von Web-Apps kann keine eindeutige Trennung dieser beiden Technologien mehr vorgenommen werden, sodass der Benutzer unter Umständen verunsichert ist, da das Verhalten nicht seinen Erwartungen entspricht.

Weiterhin ist eine Web-App nicht konsistent mit dem jeweiligen System gestaltet. Ein Layout wird für alle Systeme angeboten, sodass der Benutzer seine Erfahrungen und Kenntnisse nicht verwenden kann.

Für die Erstellung solcher Web-Apps können unter anderem die Frameworks jQuery Mobile und Sencha Touch verwendet werden. jQuery Mobile wird in Kapitel 3.3 beispielhaft vorgestellt.

PhoneGap ermöglicht ebenfalls die Erstellung einer Web-App, diese kann aber auch für unter anderem iOS, Android und BlackBerry exportiert werden. Auf diesem Weg wird eine einzige Anwendung entwickelt und trotzdem für mehrere Systeme bereit gestellt.¹⁸¹ Details zum Open Source Projekt PhoneGap werden in Kapitel 3.4 besprochen.

Grundsätzlich muss bei jedem neuen Projekt entschieden werden, welche Technologie verwendet werden soll. Der wichtigste Punkt an dieser Stelle ist der Benutzer, seine Bedürfnisse und Fähigkeiten sowie die eigentliche Aufgabe, die dieser mit der Anwendung lösen soll. Natürlich können auch die Kosten für das jeweilige Unternehmen nicht außer Acht gelassen werden. In den meisten Fällen, bei denen die Unterstützung von mobilen Geräten nur eine Ergänzung zum Angebot des Unternehmens ist, wird eine mobile Webseite oder Web-App ausreichen. Die Erstellung ist vergleichsweise einfach und kostengünstiger. Richtet sich das Ziel der Firma allerdings auf die optimale Unterstützung von mobilen Geräten mit der Ausnutzung aller verfügbaren Funktionen, ist die Verwendung von nativen Apps sinnvoll.

178 Siehe W3C: Geolocation API Specification, 10.5.2012

179 Vgl. Fling 2009, S. 147f

180 Vgl. Apple: iOS Human Interface Guidelines, Platform Characteristics

181 Vgl. Fling 2009, S. 150

3.3 Technologien für das mobile Web

Um eine Technologie für die Umsetzung auszuwählen, muss zunächst entschieden werden, wie die mobile Version der Webseite ausgeliefert werden soll. Beim mobilen Web wird zwischen einer separaten und optimierten Webseite sowie einer Webseite für alle Geräte unterschieden. In den folgenden Kapiteln werden Verwendung sowie Vor- und Nachteile dieser beiden Möglichkeiten aufgeführt.

3.3.1 Separate mobile Webseite

Eine separate mobile Seite wird speziell für den mobilen Nutzungskontext zusammengestellt und somit in Aussehen sowie Informations- und Funktionsumfang an diesen angepasst. Das bedeutet ein Reduzieren der Funktionen und des Inhalts sowie das Vergrößern von Interaktionselementen um die Bedienung mit den Fingern besser zu unterstützen.¹⁸² Nielsen hebt mit dem folgenden Zitat die Notwendigkeit der Erstellung von zwei separaten Seiten hervor.

„Good mobile user experience requires a different design than what’s needed to satisfy desktop users. Two designs, two sites, and cross-linking to make it all work.“¹⁸³

Nielsen empfiehlt weiterhin, beide Seiten miteinander zu verlinken. Findet der Benutzer innerhalb der reduzierten mobilen Version nicht die Informationen, die er benötigt, kann er auf diese Weise zur kompletten Seite gelangen. Wird der Benutzer zum Beispiel nach einer Suche mit seinem mobilen Gerät trotzdem auf die vollständige Seite geleitet, kann er mit Hilfe des Links leicht zur mobilen Version wechseln.¹⁸⁴

Da die Bildschirmgröße bei mobilen Geräten sehr unterschiedlich ist, schlägt Nielsen sogar das Anbieten einer dritten Version der Webseite vor. Auf Tablets wie dem iPad funktionieren vollständige Webseiten gut. Auf kleineren Tablets wie dem Kindle hingegen ist die komplette Webseite zu umfangreich und sollte für diese Größe erneut angepasst werden.¹⁸⁵

Der größte Vorteil einer separaten mobilen Webseite ist die optimale Anpassung an den mobilen Nutzungskontext und die Bedürfnisse der Benutzer. Auf diese Weise entstehen optimale Erlebnisse für die Nutzer von Rechnern mit großen Bildschirmen und für die Nutzer von mobilen Geräten mit tendenziell kleineren Displays.

Weiterhin kann die Reihenfolge der Inhalte und somit die benötigte Struktur der mobilen Seite von der normalen Webseite unter Umständen abweichen, zum Beispiel bei der Navigation.¹⁸⁶ Normalerweise ist die Navigation oben angeordnet und nimmt zusammen mit dem Kopfbereich der Seite einen großen Bereich ein. Dies ist für die mobile Benutzung nicht optimal, denn auf kleinen Bildschirmen nehmen diese Elemente zu viel Platz weg. Der Benutzer sollte hingegen bereits ohne Scrollen den Inhalt der Seite sehen. Es kann daher sinnvoll sein, die

182 Vgl. Nielsen: Mobile Site vs. Full Site, 10.4.2012

183 Nielsen: Mobile Site vs. Full Site, 10.4.2012

184 Vgl. Nielsen: Mobile Site vs. Full Site, 10.4.2012

185 Vgl. Nielsen: Mobile Site vs. Full Site, 10.4.2012

186 Vgl. Wroblewski: Why Seperate Mobile & Desktop Web Pages?, 1.9.2011

Navigation unten anzuordnen. An dieser Stelle muss eine unterschiedliche HTML-Struktur ausgeliefert werden. Mit Hilfe einer separaten Seite kann dieser Anspruch erfüllt werden.¹⁸⁷

Die Auslieferung von reduzierten Inhalten verbessert die Performance der Seite. Dies ist besonders bei langsamer oder auch teurer Internetverbindung wichtig. Vor allem Medien wie Bilder und Videos müssen an den mobilen Kontext angepasst und entsprechend verkleinert werden. Es gibt zwar die Möglichkeit, Medien an die Größe des Endgerätes anzupassen, allerdings werden trotzdem die großen Bilder heruntergeladen und im Anschluss zum Beispiel per CSS verkleinert. Alternativ ist es möglich, die Medien bereits serverseitig oder mit Javascript entsprechend zu skalieren und erst dann an den Benutzer auszuliefern.¹⁸⁸

Entscheidend ist dabei, jedem Gerät nur das anzubieten, was für dieses auch wirklich benötigt wird. Für mobile Endgeräte sollten reduzierte Medien und eine entsprechend angepasste und unter Umständen auch vereinfachte HTML-Struktur angeboten werden.

Weiterhin sollten auch die erweiterten Funktionen der mobilen Geräte genutzt werden. Dies können zum Beispiel die Kamera oder die GPS-Lokalisierung sein.

Der erhöhte Pflegeaufwand steht als schwerwiegender Nachteil den Vorteilen gegenüber. Sollen neue Inhalte oder sogar neue Funktionen eingepflegt werden, vervielfacht sich der Aufwand, da diese Aufgaben für zwei oder sogar mehr separate Seiten durchgeführt werden müssen. Nur wenige Unternehmen können diese zusätzlichen Aufwände finanziell und zeitlich erbringen. Außerdem erscheinen in relativ kurzen Abständen immer neue Geräte mit abweichenden Auflösungen oder Bildschirm-Abmessungen. Dadurch ist eine Anpassung an bestimmte Größen oder Geräte nicht mehr sinnvoll.¹⁸⁹ Wird hingegen nur eine Seite eingesetzt, müssen in den meisten Fällen auch nur Layout-Anpassungen vorgenommen werden. Die Verwendung sowie Vorteile und Nachteile einer einzigen Webseite, die für alle Geräte optimiert ist, werden im folgenden Kapitel erläutert.

3.3.2 Eine Webseite für alle Geräte

Für alle Benutzer, egal mit welchem Gerät diese eine Webseite aufrufen, sollten die gleichen Informationen und Möglichkeiten angeboten werden. Benutzer, die mit mobilen Endgeräten eine Internetseite aufrufen, erwarten meist den gleichen Informationsumfang und weitestgehend die gleichen Funktionen die auch eine Desktop-Seite bietet.¹⁹⁰ Dies hebt auch das folgende Zitat hervor.

„Just because I’m on a small screen doesn’t mean I’m interested in less content or want to do less.“¹⁹¹

187 Vgl. Wroblewski: Why Seperate Mobile & Desktop Web Pages?, 1.9.2011

188 Vgl. Wroblewski: Why Seperate Mobile & Desktop Web Pages?, 1.9.2011

189 Vgl. Clark: Nielsen is wrong on mobile, 12.4.2012

190 Vgl. Clark: Nielsen is wrong on mobile, 12.4.2012

191 Clark: Nielsen is wrong on mobile, 12.4.2012

Weiterhin ist zu beachten, dass viele Benutzer das Internet zunehmend fast ausschließlich mit einem mobilen Endgerät aufrufen.¹⁹² Dies verdeutlicht die Wichtigkeit, alle Informationen gleichermaßen auf allen Geräten darzustellen.

Ein weiteres Problem einer separaten Seite im Gegensatz zu einer einzigen ist die nicht korrekt funktionierende Google Suche. Ist der gesuchte Inhalt nur auf der Desktop-Webseite zu finden, landet der Benutzer auf der Homepage der mobilen Seite und muss nun die benötigten Inhalte selbst suchen.¹⁹³

Auch das World Wide Web Consortium unterstützt den sogenannten Ansatz „One Web“:

„One Web means making, as far as is reasonable, the same information and services available to users irrespective of the device they are using. However, it does not mean that exactly the same information is available in exactly the same representation across all devices. The context of mobile use, device capability variations, bandwidth issues and mobile network capabilities all affect the representation.“¹⁹⁴

Beim Konzept „One Web“ geht es darum, jedem Nutzer das bestmögliche Erlebnis anzubieten, egal welches Gerät er nutzt oder welche Fähigkeiten er besitzt. Trotzdem sollten die abweichenden Möglichkeiten der mobilen Endgeräte, wie zum Beispiel die Nutzung der Kamera oder die Ortung über GPS beachtet werden.

Neben den genannten Vorteilen hat eine einzige Webseite für alle Geräte zu erstellen auch Konsequenzen. Eine Anpassung der HTML-Struktur oder des Funktionsumfangs, wie im vorherigen Kapitel beschrieben, ist auf diesem Weg nicht möglich. Dadurch wird auch unnötiger Code zum Benutzer übertragen, sodass die Webseite unter Umständen langsam lädt oder hohe Kosten verursacht. Funktionen, die mobile Geräte ausmachen, wie die Nutzung der Kamera und die Ortung über GPS können der Seite nicht mehr hinzugefügt werden.

Um von den Vorteilen einer separaten Seite sowie einer einzigen Webseite, die für alle Geräte optimiert ist zu profitieren, bietet sich eine Kombination an. Das Konzept Responsive Web Design with Server Side Components (RESS) kann zu diesem Zweck eingesetzt werden. Diese Technologie wird in Kapitel 3.3.2.4 erläutert.

In den folgenden Unterkapiteln sind Technologien beschrieben, mit deren Hilfe das Konzept „One Web“ umgesetzt werden kann. Dabei handelt es sich um „Responsive Webdesign“ und „Mobile First“.

3.3.2.1 Responsive Webdesign

Der Begriff Responsive Webdesign wurde 2011 von Ethan Marcotte in seinem gleichnamigen Buch geprägt.¹⁹⁵ Bei dieser Herangehensweise wird eine einzige Webseite über Anpassungen des Layouts für alle Geräte optimiert.

In Abbildung 27 ist eine Webseite zu sehen, die durch Responsive Design für verschiedene Bildschirmgrößen optimiert wurde. Die Position der Container und die Größe der Bilder passt

¹⁹² Vgl. Clark: Nielsen is wrong on mobile, 12.4.2012

¹⁹³ Vgl. Combrinck: Designers respond to Nielsen on mobile, 12.4.2012

¹⁹⁴ W3C: Mobile Web Best Practices 1.0, Basic Guidelines

¹⁹⁵ Marcotte, Ethan: Responsive Web Design 2011

sich entsprechend des verfügbaren Platzes an. Auf diese Weise kann die Webseite sowohl auf einem herkömmlichen Computer, als auch auf einem Tablet und Smartphone effektiv genutzt werden.

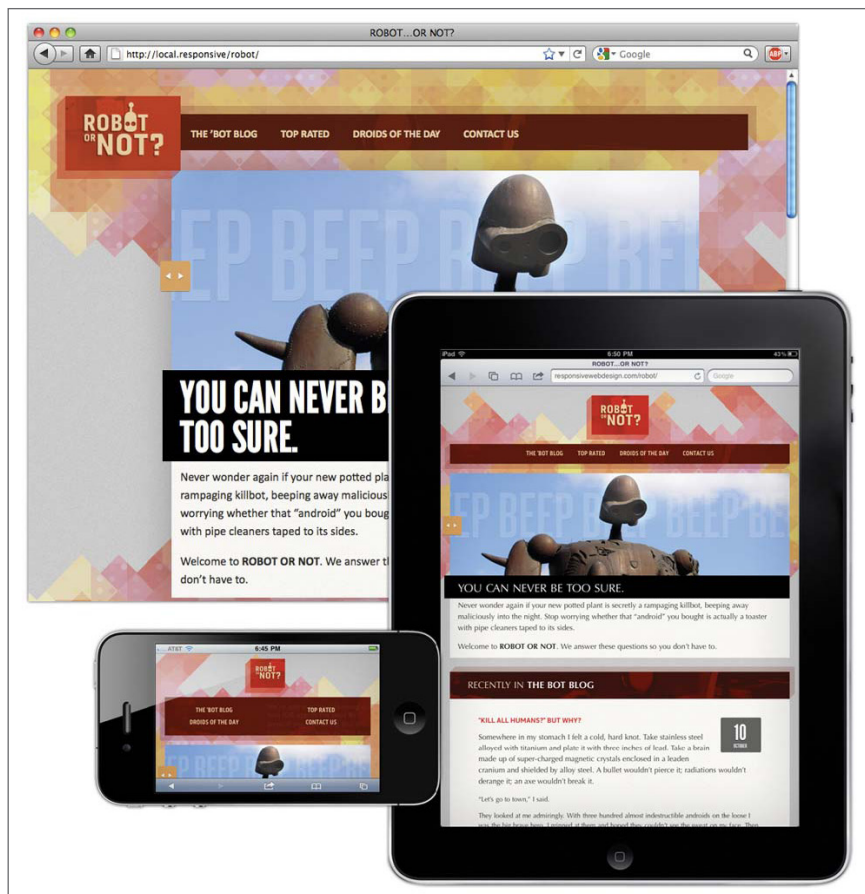


Abbildung 27 Beispiel Responsive Webdesign¹⁹⁶

Responsive Design beinhaltet ein flexibles Layout, welches sich an einem bestimmten Raster ausrichtet und an unterschiedliche Bildschirmgrößen anpasst.¹⁹⁷ Um das Layout entsprechend flexibel zu halten sind statt absoluten Maßen, wie zum Beispiel Pixel die relativen Maße Prozent und „em“¹⁹⁸ zu nutzen. Dadurch passen sich die Elemente dynamisch an die Größe des verfügbaren Bereiches an. Für die Einteilung der Seite in Bereiche ist die Verwendung von Prozent-Angaben üblich, bei Schriftgrößen wird hingegen auf „em“ zurück gegriffen.¹⁹⁹

Auch Medien können entsprechend flexibel gestaltet werden. Zu dem Zweck werden diese Medien mit der Größe des umschließenden Containers skaliert. Die Breite der Medien wird auf 100% gesetzt, sodass diese den Container immer vollständig ausfüllen.²⁰⁰ Bei einer

¹⁹⁶ Marcotte 2011, S. 95

¹⁹⁷ Vgl. Marcotte 2011, S. 9

¹⁹⁸ em: Relativ zur Größe der Schrift. 1em entspricht der aktuellen Schriftgröße, zum Beispiel 12pt. 2em sind in diesem Zusammenhang 24pt. Vgl. W3C: Web Style Sheets - CSS tips & tricks, Units, 4.4.2012

¹⁹⁹ Vgl. Marcotte 2011, S. 20, 29

²⁰⁰ Vgl. Marcotte 2011, S. 45f

Verringerung der Bildschirm-Breite wird gleichzeitig die Breite des Containers und somit die des Mediums angepasst.

Für die Umsetzung des flexiblen Layouts und somit der Technik Responsive Design sind CSS3 Media Queries notwendig. Bei CSS handelt es sich um Anweisungen, welche die Darstellung von HTML-Dateien und somit von Webseiten beeinflussen. Mit Media Queries ist es nun möglich, diese Anweisungen nur ab einer bestimmten Breite oder Höhe des Bildschirms anzuwenden. In Abbildung 28 ist ein Beispiel für den Einsatz von Media Queries zu sehen. Dem Bereich mit der Klasse `content` wird erst ab einer Breite von 400 Pixeln die Schriftgröße von 70% zugewiesen.

```
@media screen and (min-width: 400px) {
  .content {font-size: 70%;}
}
```

Abbildung 28 Media Queries²⁰¹

Media Queries können nicht nur auf die Breite oder Höhe angewendet werden, sondern auch auf weitere Eigenschaften. In Tabelle 10 sind die wichtigsten aufgeführt und kurz erläutert.

Eigenschaft	Beschreibung
width	Breite des Display-Bereiches - Breite des Viewports inklusive Scroll-Balken min-width und max-width können genutzt werden
height	Höhe des Display-Bereiches - Höhe des Viewports inklusive Scroll-Balken min-height und max-height können genutzt werden
device-width	Breite des Gerätes min-device-width und max-device-width können genutzt werden
device-height	Höhe des Gerätes min-device-height und max-device-height können genutzt werden
orientation	Orientierung des Gerätes, Portrait oder Landscape
aspect-ratio	Verhältnis von Breite zu Höhe des Display-Bereiches Zum Beispiel 16:9
device-aspect-ratio	Verhältnis von Breite zu Höhe des Gerätes
resolution	Auflösung (zum Beispiel 72dpi)

Tabelle 10 Eigenschaften für Media Queries²⁰²

²⁰¹ Vgl. W3C: Media Queries, 19.6.2012

²⁰² Vgl. Marcotte 2011, S. 76ff
Vgl. W3C: Media Queries, 19.6.2012

Media Queries können per `and` miteinander verknüpft werden, sodass es möglich ist, verschiedene Funktionen zu kombinieren.

Meist wird für Media Queries die Breite genutzt, wie auch in Abbildung 29 zu sehen. Es gibt zahlreiche bereits vorgefertigte Media Queries für verbreitete Geräte, zum Beispiel für das iPhone oder iPad.²⁰³ Der folgende Ausschnitt ist einer solchen Vorlage entnommen.

```
/* Smartphones (portrait) ----- */
@media only screen
and (max-width : 320px) {
/* Styles */
}

/* iPads (portrait and landscape) ----- */
@media only screen
and (min-device-width : 768px)
and (max-device-width : 1024px) {
/* Styles */
}
```

Abbildung 29 Media Queries für Standard Geräte²⁰⁴

Bei der Verwendung von Responsive Webdesign geht es vor allem darum, dem Benutzer die bestmögliche Erfahrung anzubieten, egal welches Gerät er benutzt. Um die Möglichkeiten dieser Technologie auszunutzen, sollte diese zukunftsorientiert verwendet werden. Die Nutzung von vorher festgelegten Media Queries, die sich zum Beispiel auf die Produkte von Apple beziehen, ist daher nicht zweckmäßig.²⁰⁵ Aus diesem Grund sollten die Werte für die Media Queries, auch Breakpoints genannt, abhängig vom Design einer Webseite individuell festgelegt werden.²⁰⁶

Zunächst sollte die Webseite oder Anwendung wie bisher gestaltet werden, also optimiert auf eine bestimmte Größe, zum Beispiel einen herkömmlichen Desktop-Computer. Es ist allerdings auch möglich, im Rahmen des mobilen Kontextes zu beginnen. Anschließend werden die Maße der Bildschirmbreite gesucht, ab denen das Layout zerstört aussieht oder die Texte nicht mehr lesbar sind.²⁰⁷ Eine einfache Möglichkeit diese Punkte ausfindig zu machen ist die Verkleinerung oder Vergrößerung des Browserfensters. Auf diese Weise können die Breakpoints für ein individuelles Layout festgelegt und dieses entsprechend korrigiert werden. Es kommt nun nicht mehr darauf an, welches Gerät mit welcher Bildschirmgröße der Benutzer verwendet, sondern die Gestaltung ist für jede beliebige Größe optimiert.²⁰⁸

Diese Herangehensweise kann auch unter dem Begriff Graceful Degradation zusammengefasst werden. Webseiten und Anwendungen werden bei diesem Konzept für die aktuellsten Browser bzw. Geräte mit dem größten Funktionsumfang erstellt. Im Anschluss findet ein

203 Vgl. Johnson: Responsive Design, Why You're Doing It Wrong, 22.2.2012

204 CSS Tricks: Media Queries for Standard Devices

205 Vgl. Johnson: Responsive Design, Why You're Doing It Wrong, 22.2.2012

206 Vgl. Johnson: Responsive Design, Why You're Doing It Wrong, 22.2.2012

207 Vgl. Johnson: Responsive Design, Why You're Doing It Wrong, 22.2.2012

208 Vgl. Johnson: Responsive Design, Why You're Doing It Wrong, 22.2.2012

Reduzieren von Inhalt und Funktionen statt, sodass auch kleine, einfache Geräte genutzt werden können.²⁰⁹

Es ist zu beachten, dass Responsive Webdesign nur das Layout betrifft und keine Anpassungen von Funktionen mit dieser Technologie vorgenommen werden können. Dies ist ein großer Nachteil, denn es können weder Funktionen reduziert noch für den mobilen Kontext sinnvoll ergänzt werden.

Dem Konzept Graceful Degradation stehen Progressive Enhancement bzw. Mobile First gegenüber, welche im folgenden Kapitel erläutert werden.

3.3.2.2 *Mobile First*

Im Rahmen des Konzepts Mobile First bzw. Progressive Enhancement wird die Webseite oder Anwendung zunächst für den mobilen Kontext optimiert. Design, Inhalt und Funktionen werden demnach auf kleine Bildschirme und geringes Leistungsvermögen sowie speziell auf die Besonderheiten von mobilen Geräten zugeschnitten. Anschließend wird der Funktionsumfang entsprechend für Computer und Laptops erhöht.²¹⁰

Mit Hilfe von Mobile First bekommt der Benutzer eines mobilen Geräts wirklich nur die Inhalte und Funktionen, die er auch nutzen kann. Es werden keine unnötigen Inhalte oder Medien übertragen, sodass die begrenzte Geschwindigkeit sowie eventuell entstehende Kosten beachtet werden.²¹¹

Außerdem erhalten Geräte, die Media Queries nicht unterstützen, auf diese Weise trotzdem die für den mobilen Kontext optimierte Seite. Beim herkömmlichen Ansatz können die Benutzer solcher Geräte nur die Desktop-Seite benutzen.²¹²

Ein weiterer Vorteil dieser Strategie ist die Reduktion auf das Wesentliche. Von der dadurch entstehenden übersichtlichen Struktur und einfachen Navigation profitieren auch die Nutzer der Desktop-Webseite.²¹³ Insgesamt wirkt die Seite aufgeräumter und der Benutzer wird nicht durch unnötige Informationen abgelenkt.

Der Ansatz Mobile First schließt Responsive Webdesign nicht aus, sondern kann mit dieser Technologie verbunden werden. Die einzige Abweichung zu Responsive Design, welches im vorherigen Kapitel vorgestellt wurde, ist der Beginn mit dem mobilen Kontext.

Mobile First ist ein in der Praxis relevanter Ansatz. Google, Facebook und Adobe nutzen diesen bereits für ihre Projekte.²¹⁴

209 Vgl. Johnson: Mobile First Design, Why It's Great and Why It Sucks, 13.3.2012
Vgl. Gustafson: Understanding Progressive Enhancement, 7.10.2008

210 Vgl. Johnson: Mobile First Design, Why It's Great and Why It Sucks, 13.3.2012

211 Vgl. Johnson: Mobile First Design, Why It's Great and Why It Sucks, 13.3.2012

212 Vgl. Jankord: Hybrid Responsive Web Design, 29.2.2012
Vgl. Marcotte 2011, S. 126

213 Vgl. Wroblewski 2011, S. 21f

214 Vgl. Wroblewski 2011, S.2

3.3.2.3 Javascript Breakpoints

Responsive Webdesign und Mobile First Responsive Webdesign lösen nur die auf das Layout bezogenen Probleme bei der Entwicklung für mobile Geräte. Funktionen können für den mobilen Kontext oder die Nutzung mit einem Computer nicht angepasst werden.

Bei der Verwendung von Javascript Breakpoints werden Punkte bestimmt, die für Anpassungen des Layouts und der Funktionen genutzt werden können.²¹⁵ In Abbildung 30 werden zum Beispiel Bereiche in Form von Pixel-Maßen definiert. Jedem Bereich wird anschließend eine CSS-Datei zugeordnet.

Das Javascript Framework „Adapt.js“ von Nathan Smith bietet sich für die Umsetzung von Javascript Breakpoints an.²¹⁶ Mit Hilfe dieses Frameworks kann definiert werden, welche CSS-Datei für welche Bildschirmgröße geladen wird. In Abbildung 30 ist die Einbindung der unterschiedlichen CSS-Dateien zu sehen.

```
var ADAPT_CONFIG = {  
  
  // Where is your CSS?  
  path: ,assets/css/',  
  
  // First range entry is the minimum.  
  // Last range entry is the maximum.  
  // Separate ranges by „to“ keyword.  
  range: [  
    ,0px      to 760px  = mobile.css`,  
    ,760px    to 980px  = 720.css`,  
    ,980px    to 1280px = 960.css`,  
    ,1280px   to 1600px = 1200.css`,  
    ,1600px   to 1920px = 1560.css`,  
    ,1940px   to 2540px = 1920.css`,  
    ,2540px                   = 2520.css`  
  ]  
};
```

Abbildung 30 Nutzung von Adapt.js²¹⁷

Der Vorteil dieses Tools ist das Laden der Datei noch vor der Auslieferung der Seite im Browser. Auf diese Weise wird nur die Datei heruntergeladen, die der Benutzer aktuell ausgeliefert bekommt.²¹⁸

Die Breakpoints, die für die CSS-Dateien genutzt werden, können ebenso für Javascript zum Einsatz kommen. Dies kann zum Beispiel bei der Navigation nützlich sein. Während für den PC optimierte Webseiten eine stets sichtbare, größere Navigation im Kopfbereich der Seite haben, sollte für mobile Geräte eine Alternative angeboten werden. Dies kann zum Beispiel eine Auswahlliste sein, die per Javascript eingefügt wird. Die Auswahlliste soll allerdings nur

²¹⁵ Vgl. Croft: How I'm implementing Responsive Web Design, 3.2.2012

²¹⁶ Vgl. Croft: How I'm implementing Responsive Web Design, 3.2.2012
Vgl. Adapt.js: Adaptive CSS

²¹⁷ Adapt.js: Adaptive CSS

²¹⁸ Vgl. Adapt.js: Adaptive CSS

für den mobilen Kontext verwendet werden, sodass an dieser Stelle ein Javascript Breakpoint genutzt werden kann.²¹⁹

3.3.2.4 RESS: Responsive Webdesign with Server Side Components

Die in diesem Kapitel vorgestellte Technologie „RESS“ vereint die Vorteile von Responsive Webdesign und einer serverseitigen Lösung. Mit Hilfe dieser Technologie ist es möglich, nur den benötigten Code zum Benutzer zu senden.²²⁰

Durch die Verwendung von Responsive Webdesign wird eine konsistente URL für die Webseite genutzt sowie basierend auf den Eigenschaften des jeweiligen Gerätes das Layout angepasst. Bei diesen Eigenschaften handelt es sich zum Beispiel um die Breite oder Höhe des Bildschirms. Allerdings können mit dieser Technologie die HTML-Struktur, Medien und Funktionen nicht angepasst werden. Durch die serverseitige Lösung ist es nun möglich, bestimmte Teile einer Webseite je nach Gerät speziell für dieses auszuliefern.²²¹

Eine Anwendungsmöglichkeit ist zum Beispiel der Austausch der Navigation nur für mobile Geräte. Wie in Abbildung 31 zu sehen ist, befindet sich das Menü auf dem mobilen Gerät unten im Fußbereich der Seite um Platz zu sparen. Auf der herkömmlichen Webseite wird die Navigation größer im Header angezeigt.

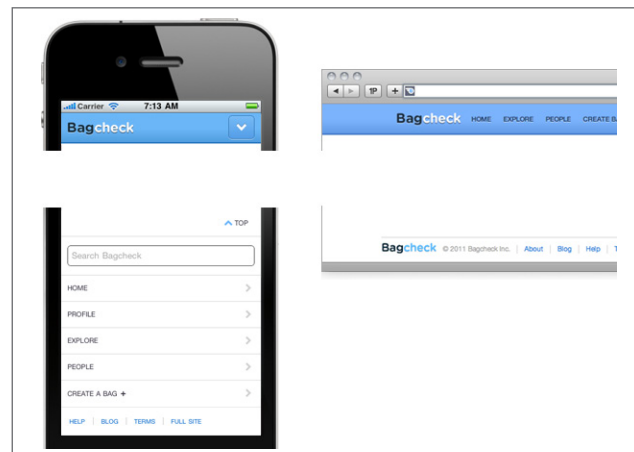


Abbildung 31 Webdesign mit RESS²²²

Um den Kopf- und Fußbereich der Seite so flexibel zu halten, müssen für diese Bereiche Platzhalter eingefügt werden. Dies kann wie in Abbildung 32 aussehen.

Zusätzlich werden je zwei HTML-Dateien für den Kopf- und Fußbereich angelegt. Die erste Datei ist der Standard für herkömmliche Webseiten, die zweite die Anpassung für die mobilen Geräte. Auf dem Server wird nun je nach Gerät die passende Datei ausgeliefert.

²¹⁹ Vgl. Croft: How I'm implementing Responsive Web Design, 3.2.2012

²²⁰ Vgl. Wroblewski: RESS: Responsive Design + Server Side Components, 12.9.2011

²²¹ Vgl. Wroblewski: RESS: Responsive Design + Server Side Components, 12.9.2011

²²² Wroblewski: RESS: Responsive Design + Server Side Components, 12.9.2011

```
<body>
{{>header}}

[...document content...]

{{>footer}}
</body>
```

Abbildung 32 Dynamischer Header und Footer²²³

Auf diese Weise können auch Medien wie zum Beispiel Bilder entsprechend angepasst zum Nutzer gesendet werden. Abbildung 33 zeigt den HTML-Code innerhalb der Datei für die normale Webseite (userimage.html) sowie für mobile Geräte (mobile_userimage.html).

```
userimage.html


mobile_userimage.html

```

Abbildung 33 Angepasste Auslieferung von Bildern²²⁴

Durch die Verwendung von zwei HTML-Dateien wie in Abbildung 33 zu sehen, werden dem Benutzer keine Bilder mehr gesendet, die er nicht benötigt. Auf diese Weise wird die vor allem für mobile Geräte wichtige Übertragungsgröße reduziert.

Durch das Hinzufügen von serverseitiger Technologie steigt aufgrund der umfangreicheren Anpassungen auch der Aufwand. Neben dem Erstellen mehrerer HTML-Dateien ist für die Umsetzung die Erkennung von Gerätetypen bzw. deren Bildschirmgröße notwendig. Entsprechend des Gerätes müssen dann HTML- und CSS-Dateien zusammengestellt werden.

3.3.3 Web-Apps

Web-Apps sind eine Kombination aus mobiler Webseite und nativer App. Diese Art von Anwendungen können über den Browser aufgerufen werden, Aussehen und Funktionsweise orientieren sich aber eher an nativen Apps.

Bei Web-Apps gibt es nur eine Ansicht, die durch Nutzerinteraktion verändert bzw. aktualisiert wird. Im Gegensatz dazu wird bei Webseiten immer eine neue Seite geladen.²²⁵ Weiterhin können Web-Apps auch im Vollbildmodus genutzt werden. Dazu ist es zum Beispiel möglich, unter iOS die Bedienelemente des Browsers Safari auszublenden.²²⁶

²²³ Wroblewski: RESS: Responsive Design + Server Side Components, 12.9.2011

²²⁴ Wroblewski: RESS: Responsive Design + Server Side Components, 12.9.2011

²²⁵ Vgl. Fling 2009, S. 205

²²⁶ Vgl. Apple: iOS Human Interface Guidelines, Platform Characteristics

Google bietet für seine E-Mail Anwendung unter anderem eine Web-App an. In Abbildung 34 ist die Übersicht über den Posteingang zu sehen. Die Anwendung erinnert in Aussehen und Funktion eher an eine native App als an eine Webseite.

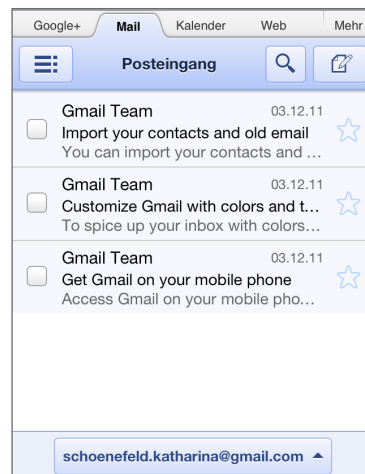


Abbildung 34 Web-App Google Mail

Für die Umsetzung von Web-Apps können Frameworks eingesetzt werden. In diesem Kapitel wird beispielhaft das Framework jQuery Mobile vorgestellt.

Bei jQuery Mobile handelt es sich um ein Framework, welches für den mobilen Kontext und die Bedienung mit einem Touchscreen optimiert ist. Das Framework baut auf dem weit verbreiteten jQuery Core und dem jQuery User Interface auf und wird unter anderem von Mozilla, Adobe und Nokia gesponsert.²²⁷

jQuery Mobile ist kompatibel mit einem Großteil der mobilen Geräte, da es auf klarem, semantischen HTML aufbaut. Das Framework bietet sogar Accessibility Funktionen, wie zum Beispiel WAI-ARIA²²⁸. Mit Hilfe dieser Technologie wird die Unterstützung von Screen-readern, zum Beispiel auch VoiceOver unter iOS, erheblich verbessert.²²⁹ WAI-ARIA wird in Kapitel 3.5.1 besprochen.

Weiterhin bietet das Framework Effekte und Übergänge an, die von nativen Apps unter iOS, Android usw. bekannt sind.²³⁰ Auf diese Weise wird beim Benutzer noch mehr das Gefühl geweckt, dass es sich um eine native App handelt.

Problematisch bei der Nutzung von jQuery Mobile ist allerdings die schwierige Kombination des Frameworks mit Responsive Design. Entweder nutzt man eine separate mobile Seite, um das Problem zu umgehen, oder man verwendet das jQuery Mobile Design auch für die Präsentation der normalen Webseite.²³¹

Für die Verwendung von jQuery Mobile sind, wie in Abbildung 35 zu sehen, eine CSS-Datei und zwei Javascript Dateien in den HTML-Code einzubinden.

²²⁷ Vgl. Hadlock 2012, S. XII

²²⁸ WAI-ARIA: Web Accessibility Initiative - Accessible Rich Internet Applications

²²⁹ Vgl. Hadlock 2012, S. XII

²³⁰ Vgl. Layon 2012, S. 143

²³¹ Vgl. Layon 2012, S. 143

```
<head>
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.1.0/
jquery.mobile-1.1.0.min.css" />

<script src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
<script src="http://code.jquery.com/mobile/1.1.0/jquery.mobile-
1.1.0.min.js"></script>
</head>
```

Abbildung 35 Einbindung von jQuery Mobile²³²

Eine weitere wichtige Angabe im `<head>` ist der Viewport. Der Viewport ist der Bereich, in den die erstellte Web-App hinein passt. Eine typische Angabe ist in Abbildung 36 zu sehen.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Abbildung 36 Metadatum Viewport²³³

Anschließend sind dem HTML-Code `data-role` Attribute hinzuzufügen, so wie in Abbildung 37 zu sehen. Diese Attribute sind nur semantische Auszeichnungen und haben für den Benutzer oder den Browser zunächst keine Bedeutung, können aber per Javascript angesprochen werden.²³⁴

In Abbildung 37 wird die komplette Seite mit `data-role="page"` ausgezeichnet, der Inhalt mit `content` sowie Kopf- und Fußbereich mit `header` bzw. `footer`.

```
<body>
  <div data-role="page">
    <div data-role="header">
      <h1>Page title</h1>
    </div>
    <div data-role="content">
      Body copy
    </div>
    <div data-role="footer">
      Copyright
    </div>
  </div>
</body>
```

Abbildung 37 Auszeichnung einer HTML-Datei mit `data-role` Attributen²³⁵

Das Framework jQuery Mobile nutzt diese Attribute und transformiert die HTML-Datei in die Web-App, die in Abbildung 38 zu sehen ist.

²³² jQuery Mobile Framework: Download

²³³ Vgl. Firtman 2012, S. 30

²³⁴ Vgl. Hadlock 2012, S. 24

²³⁵ Hadlock 2012, S. 33

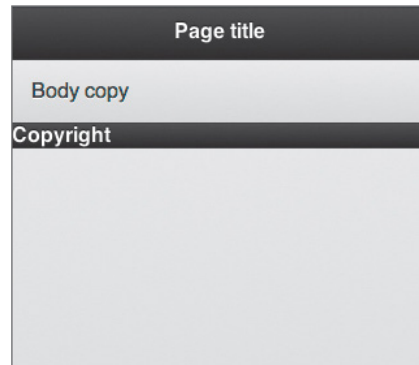


Abbildung 38 jQuery Mobile im Einsatz²³⁶

Um die Web-App zu strukturieren werden wie bei einer herkömmlichen Webseite separate HTML-Seiten erstellt. Das Element mit der Rolle `page` wird als Container genutzt. In diesem Container werden dann die anderen Seiten dynamisch geladen, wenn der Benutzer einen Link aktiviert.²³⁷

3.4 Technologien zur Erstellung von nativen Apps

Zur Erstellung von nativen Apps gibt es grundsätzlich zwei Möglichkeiten. Zunächst kann für jedes System, das unterstützt werden soll, eine eigene Anwendung entwickelt werden. Dazu sind unterschiedliche Programme sowie verschiedene Programmiersprachen notwendig.

Eine andere Möglichkeit ist die Verwendung von Frameworks wie PhoneGap. Mit Hilfe von PhoneGap kann zum Beispiel eine Anwendung erstellt und anschließend für mehrere Betriebssysteme exportiert werden. Beide Herangehensweisen werden in den folgenden Kapiteln vorgestellt.

3.4.1 Entwicklung herstellerspezifischer Apps

Für die Entwicklung von nativen Apps für iOS, Android und Windows Phone stellen die jeweiligen Hersteller eine eigene Entwicklungsumgebung zur Verfügung. Demnach müssen für diese Systeme drei unterschiedliche Programme sowie auch drei verschiedene Programmiersprachen verwendet werden.

Apps für das Betriebssystem iOS werden mit dem Programm Xcode²³⁸ von Apple in der Sprache Objective C erstellt. Xcode beinhaltet neben der Entwicklungsumgebung auch Werkzeuge zum grafischen Zusammenstellen der Benutzeroberfläche sowie zur Performance-Optimierung. Mit dem iOS Simulator können die entwickelten Apps direkt getestet werden.²³⁹

²³⁶ Hadlock 2012, S. 33

²³⁷ Vgl. Hadlock 2012, S. 36

²³⁸ Apple Developer: Developer Tools

²³⁹ Vgl. Apple Developer: Developer Tools

Für die Entwicklung von Android Apps benötigt man zunächst das Android SDK. Zusätzlich sollte das Plugin „Android Developer Tools“ für Eclipse verwendet werden, um mit dieser Software die Apps programmieren zu können. Mit Hilfe des „Android Virtual Device Managers“ können verschiedene Endgeräte hinzugefügt werden, um die erstellten Apps über den Emulator zu testen.²⁴⁰ Android Apps werden mit Hilfe von HTML, CSS und Java erstellt.

Auch die Entwicklung für Windows Phone 7 wird über eine individuelle Software durchgeführt. An dieser Stelle ist Microsoft Visual Studio in Verbindung mit den „Windows Phone Developer Tools“ zu verwenden.²⁴¹ Für die Entwicklung von Windows Phone Apps sind Kenntnisse in einer dritten Programmiersprache, C#, notwendig.

Die Erstellung von nativen Apps ist sehr aufwendig, denn meist soll mehr als nur ein System unterstützt werden, damit möglichst viele Benutzer die App verwenden können. Zusätzliche Kenntnisse und finanzielle sowie zeitliche Ressourcen sind notwendig, um dies umzusetzen. Für den Benutzer ist die Verwendung von nativen Apps, die speziell für das jeweilige Betriebssystem entwickelt wurden, von Vorteil. Die Konsistenz mit dem Betriebssystem hilft dem Benutzer bei der Verwendung der App, da er seine erlangten Kenntnisse und Erfahrungen einsetzen kann.

Im nächsten Kapitel wird eine Technologie vorgestellt, die einen Kompromiss bietet. Mit geringerem Aufwand können mit Hilfe von PhoneGap native Apps für unterschiedliche Systeme erstellt werden.

3.4.2 Herstellerunabhängige Erstellung mit PhoneGap

Bei PhoneGap handelt es sich um ein Open Source Framework. Mit diesem können Web-Apps mit den Technologien HTML, CSS und Javascript erstellt werden. Anschließend besteht die Möglichkeit, native Apps für unter anderem iOS, Android und Windows Phone zu exportieren und in den jeweiligen App Store zu stellen. Auf diese Weise wird nur eine App erstellt, die dann auf viele Systeme verteilt werden kann.²⁴² Diese Art von Apps werden auch hybride Apps genannt, da sie Technologien für das Web und für native Anwendungen verbinden.

Mit Hilfe von PhoneGap kann, im Gegensatz zu Web-Apps, ein Großteil der nativen Funktionen der mobilen Geräte genutzt werden. Dies betrifft unter anderem die Kamera, die Kontakte des Nutzers, die gespeicherten Daten, die aktuelle Position sowie die für das jeweilige System spezifischen Meldungen.

PhoneGap nutzt für die nativen Apps die Web-Ansicht des jeweiligen Systems. Unter iOS ist das zum Beispiel die UIWebView, die zum Darstellen von Web-Inhalten verwendet wird.²⁴³ Die native App beinhaltet somit eine Ansicht der Web-App wie im Browser, nur ohne dessen Bedienelemente. Andere, systemeigene Elemente werden an dieser Stelle nicht genutzt. Die mit PhoneGap erstellten Apps sehen somit auf allen Geräten gleich aus und funktionieren ebenfalls auf die gleiche Art und Weise.

240 Vgl. Android Developer: Develop, Android Training

241 Vgl. Microsoft: Windows Phone, Schritt für Schritt zur eigenen App

242 Vgl. Adobe Systems Inc.: PhoneGap, About the Project

243 Vgl. Adobe Systems Inc.: PhoneGap, PhoneGap Explained Visually, 2.5.2012

Da die App auf allen Geräten und auch im Browser die gleiche Gestaltung aufweist, ist die Konsistenz zum jeweiligen System nicht mehr vorhanden. Aus diesem Grund geht für den Benutzer das Gefühl verloren, dass er mit einer nativen App arbeitet. Er kann seine Erfahrungen und Kenntnisse nicht mehr anwenden und muss gegebenenfalls neue Paradigmen erlernen. In diesem Fall ist es sogar besser, bei der Web-App zu bleiben. Denn diese ruft der Benutzer über einen Browser auf und erwartet an dieser Stelle keine Konsistenz mit seinem System. Weiterhin kann der Nutzer die erlernten Paradigmen aus dem World Wide Web auf die App anwenden.

Ein weiteres Problem bei der Verwendung von hybriden Apps ist die unter Umständen mangelnde Zugänglichkeit. Unter iOS und Android müssen zum Beispiel die Standardelemente verwendet werden, damit der Benutzer einen Screenreader effektiv einsetzen kann. Für alle benutzerdefinierten Elemente müssen zusätzlich über den Quellcode Accessibility-Informationen hinzugefügt werden, damit der Screenreader die Inhalte korrekt ausgeben kann. Da diese Informationen für jedes System individuell sind, funktioniert dies nicht mit Hilfe des Frameworks PhoneGap. Die Zugänglichkeit von nativen Anwendungen für iOS und Android wird in Kapitel 3.5.2 ausführlicher behandelt.

3.5 Technologien für die Zugänglichkeit

3.5.1 WAI-ARIA

Die Accessible Rich Internet Applications Suite (ARIA) der Web Accessibility Initiative (WAI) ist eine Technologie zur semantischen Auszeichnung von HTML-Elementen.²⁴⁴ Dem Benutzer kann mit Hilfe von ARIA die Bedeutung von Elementen übermittelt werden, ohne dass dieser sie sehen muss. Somit kann er zum Beispiel zwischen Navigation und Hauptinhalt, sowie zwischen einer reinen Information und einer Interaktionsmöglichkeit unterscheiden. Dadurch wird die Zugänglichkeit dynamischer Webseiten für behinderte Menschen entscheidend verbessert.

Da es sich bei ARIA nur um eine semantische Auszeichnung handelt, bleiben Layout und Funktion unbeeinflusst. Die Verwendung von ARIA hat somit keine negativen Nebeneffekte.²⁴⁵

3.5.1.1 Document Landmark Roles

Mit Hilfe der Document Landmark Roles können HTML-Elemente mit einer semantischen Bedeutung versehen werden. Assistive Technologien geben diese Informationen an den Benutzer weiter, sodass dieser zum Beispiel die Navigation vom Hauptinhalt unterscheiden kann. Tabelle 11 führt die verfügbaren Rollen auf.

²⁴⁴ Vgl. W3C: Web Accessibility Initiative, WAI-ARIA FAQ
Vgl. Featherstone: Real World Accessibility, HTML5, ARIA and the Modern Web, 5.7.2011

²⁴⁵ Vgl. W3C: Web Accessibility Initiative, WAI-ARIA FAQ

Rolle	Beschreibung
article	Inhalt mit eigenständigem Sinn
banner	Spezieller Inhalt für die gesamte Webseite (Titel der Seite, Logo)
complementary	Unterstützende Informationen zum Hauptinhalt
contentinfo	Footerbereich (Fußnoten, Copyright, rechtliche Hinweise)
main	Hauptinhalt
navigation	Hauptnavigation, Liste von Links
search	Suchfunktion der Webseite

Tabelle 11 WAI-ARIA: Document Landmark Roles²⁴⁶

Die verschiedenen Rollen können mit dem Attribut `role` einem HTML-Element hinzugefügt werden. In Abbildung 39 ist zum Beispiel die Auszeichnung des Hauptinhaltes mit der Rolle `main` zu sehen.

```
<div id="content" role="main"> Hauptinhalt </div>
```

Abbildung 39 WAI-ARIA: Verwendung der Document Landmark Roles

3.5.1.2 Zustände und Eigenschaften

Mit Hilfe der in Tabelle 12 aufgeführten Attribute können dem Benutzer weitere Informationen zu einem Element angeboten werden. Diese Informationen können statisch sein und zum Beispiel eine Verbindung mit einer Beschriftung oder einer Beschreibung anbieten. Dies ist bei den Attributen `aria-labelledby` und `aria-describedby` der Fall. Zum anderen können die Attribute auch dynamische Eigenschaften verdeutlichen, zum Beispiel ob eine Checkbox ausgewählt ist oder nicht. In diesem Fall kann `aria-checked` verwendet werden.

Tabelle 12 zeigt eine Auswahl der in ARIA verfügbaren Attribute. Eine vollständige Übersicht kann der Spezifikation des W3C entnommen werden.²⁴⁷

Eigenschaft	Beschreibung
aria-labelledby	Identifiziert das Element, welches die Beschriftung enthält
aria-describedby	Identifiziert das Element, welches eine zusätzliche Beschreibung anbietet
aria-required	Pflichtfeld, Eingabe des Nutzers ist erforderlich
aria-invalid	Fehlerhaft ausgefülltes Feld

Tabelle 12 WAI-ARIA: Ausgewählte Zustände und Eigenschaften²⁴⁸

²⁴⁶ Vgl. W3C: Accessible Rich Internet Applications (WAI-ARIA) 1.0, 18.1.2011

²⁴⁷ W3C: Accessible Rich Internet Applications (WAI-ARIA) 1.0, 18.1.2011

²⁴⁸ Vgl. W3C: Accessible Rich Internet Applications (WAI-ARIA) 1.0, 18.1.2011

In Abbildung 40 ist beispielhaft die Nutzung der Attribute `aria-labelledby` und `aria-describedby` zu sehen. Das Label und die Beschreibung werden jeweils mit einer ID versehen. Über die beiden ARIA-Attribute werden diese IDs dann genutzt, um die Verbindung mit dem Eingabefeld herzustellen.

```
<label id="username" for="user_input">Benutzername</label>

<input id="user_input" type="text" aria-labelledby="username"
aria-describedby="user_desc" />

<span id="user_desc">Der Benutzername ist zur Identifizierung im System
notwendig</span>
```

Abbildung 40 WAI-ARIA: Verwendung von `aria-labelledby` und `aria-describedby`

3.5.1.3 Live Regions

Besonders wichtig für Benutzer von assistiven Technologien ist das Hinweisen auf Änderungen des Inhaltes. Solche dynamischen Anpassungen sind für sehende Benutzer kein Problem, da die Änderungen sichtbar sind. ARIA bietet für diese Fälle die sogenannten Live Regions an. Mit Hilfe dieser Attribute wird der Benutzer auf Änderungen aufmerksam gemacht. Die Position des Benutzers sowie der Fokus werden dabei aber nicht verändert.²⁴⁹

In Tabelle 13 sind die Live Regions und deren Werte aufgeführt und kurz beschrieben.

Attribute	Beschreibung	Wert	Beschreibung
aria-live	Änderungswahrscheinlichkeit einer Region	off	Keine Änderung
		polite	Nutzer kann aktuelle Aktivität zu Ende führen, Antwort nicht nötig
		assertive	Höhere Priorität, unterbricht Nutzer aber nicht sofort
aria-atomic	Nutzung im Zusammenspiel mit aria-live bei Änderungen des Inhaltes	true	Komplette geänderte Region wird ausgegeben
		false	Nur der geänderte Teil wird ausgegeben
aria-busy	Verhindert, dass assistive Technologien Änderungen melden, bevor Aktualisierung abgeschlossen ist	true	Abwarten, bis alle Elemente geladen sind
		false	Änderung kann gemeldet werden

²⁴⁹ Vgl. W3C: Accessible Rich Internet Applications (WAI-ARIA) 1.0, 18.1.2011

Attribute	Beschreibung	Wert	Beschreibung
aria-relevant	Gibt die Art der Änderungen einer Region an	additions	Hinzufügen von Knoten im DOM
		removals	Entfernen von Knoten aus dem DOM
		text	Text wurde hinzugefügt oder entfernt
		all	Alles gilt

Tabelle 13 WAI-ARIA: Live Regions²⁵⁰

3.5.2 Zugänglichkeit von nativen Apps

3.5.2.1 Zugänglichkeit unter iOS

Im Betriebssystem von Apple kann der Benutzer zahlreiche Einstellungen vornehmen um Inhalte alternativ ausgeben zu lassen. Neben Funktionen wie Zoom, einer Vergrößerung des Textes oder der Darstellung der Oberfläche in Schwarz / Weiß steht mit VoiceOver auch ein Screenreader zur Verfügung.²⁵¹

VoiceOver liest die Elemente auf dem Bildschirm und deren Zweck nicht nur vor, sondern ermöglicht dem Benutzer auch eine alternative Navigation. Der Benutzer hat die Möglichkeit, Elemente mit einem Tap frei auszuwählen. Das gewählte Element wird anschließend von VoiceOver vorgelesen. Mit einem doppelten Tap an einer beliebigen Stelle auf dem Bildschirm kann dieses Element nun aktiviert werden. Fällt es dem Benutzer schwer, die Elemente auf dem Bildschirm zu lokalisieren, kann er außerdem die Elemente nacheinander durchgehen. Dies wird durch ein Wischen an einer beliebigen Stelle nach links oder rechts für nächstes oder voriges Element realisiert.

Für die effektive Anwendung von VoiceOver müssen alle Elemente mit denen der Benutzer interagieren kann zugänglich sein. Diese Elemente sollten demnach korrekte und hilfreiche Informationen über deren Position auf dem Bildschirm, die Bezeichnung, das Verhalten, den Wert und den Typ geben. Werden Standardelemente und -ansichten genutzt, sind die Accessibility Informationen bereits integriert und müssen nicht extra hinzugefügt werden.²⁵² Zu beachten ist dabei allerdings, dass diese Elemente auch entsprechend ihres Zwecks und ihrer Funktion genutzt werden. Sollten doch Abweichungen notwendig sein, müssen die Informationen angepasst werden.

In Abbildung 41 sind die Eigenschaften der Elemente, die später vom Screenreader VoiceOver vorgelesen werden, zu sehen.

²⁵⁰ Vgl. W3C: Accessible Rich Internet Applications (WAI-ARIA) 1.0, 18.1.2011

²⁵¹ Vgl. Apple: Accessibility Programming Guide for iOS, Accessibility on iPhone

²⁵² Vgl. Apple: Accessibility Programming Guide for iOS, Accessibility on iPhone

Attribut	Beschreibung	Beispiel
Label	Wort oder kurze Wortgruppe zur Beschreibung des Kontrollelements oder der Ansicht	„Add“, „Play“, „Search“
Traits	Vorgegebene Eigenschaften, die das Element charakterisieren (Kombination möglich)	Button Link Search Field Keyboard Key Static Text Image Plays Sound Selected Summary Element Updates Frequently Not Enabled None
Hint	Kurze Beschreibung des Resultats einer Aktion oder eines Elements	„Adds a title“, „Opens the shopping list“
Frame	Bildschirmkoordinaten, gibt Position des Elementes an	
Value	Aktueller Wert des Elements, falls Label diesen nicht bereits beschreibt	Slider: „50%“

Tabelle 14 Attribute für die Auszeichnung von Elementen unter iOS²⁵⁶

Hinweise sind optional und sollten nur angeboten werden, wenn die Beschreibung durch das Label nicht ausreicht. Man sollte auch beachten, dass der Benutzer das Vorlesen der Hinweise deaktivieren kann. Daher ist besonders auf die Aussagekraft des Labels zu achten. Auch der Hinweis sollte nur knapp formuliert werden, da dieser immer erst vorgelesen wird, bevor der Benutzer eine Aktion ausführen kann. Ansonsten wird die Nutzung der Anwendung unnötig verzögert und der Benutzer in der Ausführung seiner Aufgabe behindert.²⁵⁷ Weiterhin sollte der Hinweis mit einem Großbuchstaben beginnen und mit einem Satzzeichen enden, um die korrekte Betonung sicher zu stellen.

Unter iOS können die Accessibility-Informationen für die Elemente über die grafische Benutzeroberfläche, wie in Abbildung 42 zu sehen, oder über den Quellcode hinzugefügt werden. Der Interface Builder bietet ein Formular an, über das Labels, Hinweise und Traits angegeben werden können. Auf diese Weise ist es möglich, ohne Kenntnisse in der Programmierung bereits wichtige Informationen für VoiceOver zu hinterlegen.

²⁵⁶ Apple: Accessibility Programming Guide for iOS, Accessibility on iPhone
Apple: Accessibility Programming Guide for iOS, Making Your iPhone Application Accessible

²⁵⁷ Vgl. Apple: Accessibility Programming Guide for iOS, Making Your iPhone Application Accessible

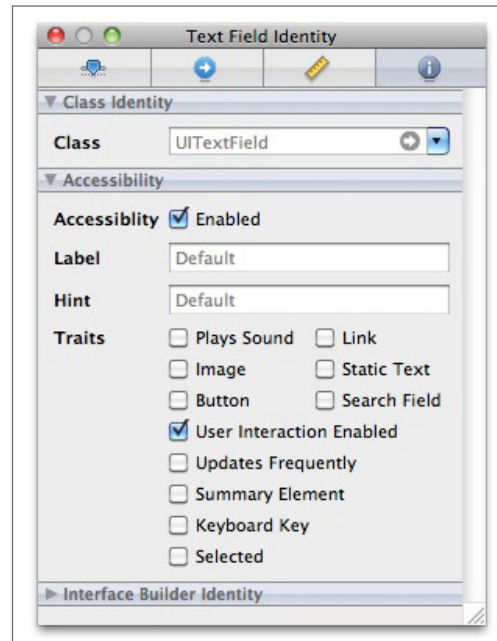


Abbildung 42 Accessibility im Interface Builder²⁵⁸

Für automatisch generierte Inhalte oder benutzerdefinierte Elemente sollten die Attribute über den Programmcode hinzugefügt werden, wie auch in Abbildung 43 zu sehen ist. Zunächst wird mit `YES` angegeben, dass es sich bei der aktuellen Ansicht um ein für den Screenreader zugängliches Element handelt. Anschließend werden die Attribute `Label`, `Traits` und `Hinweis` gesetzt. Bei dem Element in Abbildung 43 handelt es sich um einen Button.

```
@implementation MyCustomView
- (BOOL)isAccessibilityElement {
    return YES;
}

- (NSString *)accessibilityLabel {
    return NSLocalizedString(@"MyCustomView.label", nil);
}

/* This custom view behaves like a button. */
- (UIAccessibilityTraits)accessibilityTraits {
    return UIAccessibilityTraitButton;
}

- (NSString *)accessibilityHint {
    return NSLocalizedString(@"MyCustomView.hint", nil);
}
@end
```

Abbildung 43 Definition von Attributen für die Zugänglichkeit unter iOS²⁵⁹

Neben nativen Apps müssen auch Webseiten, die zum Beispiel über den Browser Safari aufgerufen werden, zugänglich sein. Für diesen Fall bietet die Technologie WAI-ARIA, die in Kapitel 3.5.1 vorgestellt wurde, geeignete Konzepte an.

²⁵⁸ Apple: Accessibility Programming Guide for iOS, Making Your iPhone Application Accessible

²⁵⁹ Apple: Accessibility Programming Guide for iOS, Making Your iPhone Application Accessible

Unter iOS kann ARIA die Benutzer von VoiceOver beim Navigieren durch Webseiten unterstützen. Eine Funktion, die im Zusammenhang mit VoiceOver aktiviert werden kann, ist der Rotor. Mit Hilfe dieses Kontrollelementes kann der Benutzer bestimmen, anhand welcher Eigenschaften der Screenreader durch die Seite navigiert. Dies können unter anderem Überschriften, Links oder Formularelemente sein.²⁶⁰

Der Benutzer kann mit zwei Fingern eine Drehung an einer beliebigen Stelle auf dem Bildschirm ausführen, um zwischen den verschiedenen Navigationsarten zu wechseln. In Abbildung 44 ist die beschriebene Geste zu sehen. Die aktuell ausgewählte Art der Navigation wird dem Benutzer vorgelesen.²⁶¹



Abbildung 44 Rotor unter iOS²⁶²

Ist eine Webseite bereits gut strukturiert und entsprechend semantisch durch HTML-Elemente ausgezeichnet, kann der Rotor effektiv eingesetzt werden. ARIA kann zusätzlich noch weitere Möglichkeiten zur Navigation hinzufügen. Der Benutzer kann zum Beispiel schneller zwischen definierten Sektionen blättern und gewinnt auf diese Weise einen besseren Überblick über die Webseite.²⁶³

Aktuell wird dem Benutzer aber nicht die Art des Elements, zum Beispiel Navigation oder Artikel vorgelesen, sondern nur „landmark start“ bzw. „Anfang des Orientierungspunkts“ und „landmark end“ bzw. „Ende des Orientierungspunkts“.²⁶⁴

3.5.2.2 Zugänglichkeit unter Android

Ebenso wie das System iOS bietet Android den Benutzern zahlreiche Funktionen zur Verbesserung der Zugänglichkeit an. Diese helfen Nutzern mit visuellen, physischen oder altersbedingten Einschränkungen bei der Bedienung des Gerätes. Ähnlich wie unter iOS sind auch die Standard-Elemente unter Android bereits zugänglich und es sind nur wenige Anpassungen notwendig.²⁶⁵ Unter Android können die notwendigen Einstellungen aber nur über den Programmcode gesetzt werden.

²⁶⁰ Vgl. Apple: Accessibility, iPhone

²⁶¹ Vgl. Gibson: Making web content more accessible with iOS VoiceOver, 6.6.2011

²⁶² Apple: Accessibility, iPhone

²⁶³ Vgl. Gibson: Making web content more accessible with iOS VoiceOver, 6.6.2011

²⁶⁴ Vgl. Gibson: Making web content more accessible with iOS VoiceOver, 6.6.2011

²⁶⁵ Vgl. Android Developer: Develop, Making Applications Accessible

Zunächst sind Labels für die Elemente zu hinterlegen, die keinen Text enthalten, wie zum Beispiel Buttons, Bilder oder Checkboxes. Dies erfolgt mit Hilfe des Attributes `android:contentDescription`.²⁶⁶ Auf diese Weise kann dem Benutzer die Bedeutung eines Elementes vermittelt werden, auch wenn er die visuelle Information, wie ein Symbol, nicht aufnehmen kann. Der Inhalt des Attributes ist nicht sichtbar, sondern wird dem Benutzer nur über den Screenreader bereitgestellt.

In Abbildung 45 ist die Verwendung des Attributes `android:contentDescription` zu sehen. Der Button wird ausschließlich durch ein Symbol dargestellt, die Information ist demnach rein visuell. Damit dem Benutzer diese Information auch über einen Screenreader angeboten werden kann, wird der String `add_note` hinterlegt. Über diese Ressource kann für die deutsche Sprache dann zum Beispiel „Notiz hinzufügen“ hinterlegt werden. Erhält das Element den Fokus, wird dem Benutzer das Label vorgelesen.

```
<ImageButton
    android:id="@+id/add_note_button"
    android:src="@drawable/add_note"
    android:contentDescription="@string/add_note"/>
```

Abbildung 45 Hinzufügen eines Labels unter Android²⁶⁷

Neben einem Label können auch Hinweise hinzugefügt werden, um dem Benutzer eine zusätzliche Beschreibung anzubieten. Dies erfolgt über das Attribut `android:hint`.

Weiterhin sollte die Richtungssteuerung aktiviert werden. Bei dieser Art der Steuerung wird es dem Benutzer ermöglicht, über Hardware, zum Beispiel einen Trackball oder Pfeil-Tasten sowie Software zu navigieren. Ziel dieser Steuerung ist es, alle Kontrollelemente auch ohne die Nutzung des Touchscreens bedienbar und zugänglich zu machen.²⁶⁸

Um diese alternative Navigation zur Verfügung zu stellen, müssen die jeweiligen Elemente zunächst fokussierbar gemacht werden. Dafür wird das Attribut `android:focusable` auf `true` gesetzt. Alternativ kann auch die Methode `setFocusable()` verwendet werden. Der Benutzer kann die Elemente nun über die Richtungssteuerung erreichen und mit ihnen interagieren, ohne den Touchscreen zu verwenden. Zu beachten ist, dass die Standard-Elemente unter Android dieses Attribut bereits besitzen und sich auch in der Gestaltung entsprechend anpassen, wenn diese den Fokus erhalten.²⁶⁹

Neben dem Setzen des Fokus kann auch die Reihenfolge individuell bestimmt werden. Eine Anpassung sollte aber nur in speziellen Fällen erfolgen, da eine Standard-Reihenfolge bereits vorgegeben ist. Diese ergibt sich aus der Anordnung der Elemente und entspricht somit im Idealfall der inhaltlichen und visuellen Reihenfolge.

Über das Setzen der Attribute `android:nextFocusDown`, `android:nextFocusLeft`, `android:nextFocusRight` und `android:nextFocusUp` kann die Standard-Reihenfolge der Elemente überschrieben werden. In Abbildung 46 ist die Verwendung der Attribute für

²⁶⁶ Vgl. Android Developer: Develop, Making Applications Accessible

²⁶⁷ Android Developer: Develop, Making Applications Accessible

²⁶⁸ Vgl. Android Developer: Develop, Making Applications Accessible

²⁶⁹ Vgl. Android Developer: Develop, Making Applications Accessible

das Setzen des Fokus zu sehen. Das nächste Element, das jeweils fokussiert werden soll, wird durch die ID, zum Beispiel über @id/text angesprochen.

```
<LinearLayout android:orientation="horizontal" ... >
  <EditText android:id="@+id/edit"
    android:nextFocusDown="@+id/text" ... />
  <TextView android:id="@+id/text"
    android:focusable="true"
    android:text="Hello, I am a focusable TextView"
    android:nextFocusUp="@+id/edit" ... />
</LinearLayout>
```

Abbildung 46 Setzen der Fokus-Reihenfolge unter Android²⁷⁰

Wenn die Standard-Elemente des Android Frameworks genutzt werden, sind keine weiteren Anpassungen für die Zugänglichkeit durchzuführen. Nur wenn benutzerdefinierte Ansichten und Elemente erstellt werden, sind zusätzlicher Aufwand und erweiterte Programmierkenntnisse erforderlich. Daher empfiehlt es sich, nur die Android Standard-Elemente zu verwenden.

²⁷⁰ Android Developer: Develop, Making Applications Accessible

4 Analyse und Bewertung ausgewählter Anwendungen

Mit Hilfe der in Kapitel 2 und 3 vorgestellten Grundlagen und Technologien wurde bereits eine wichtige Basis für den Entwurf und die Erstellung von Webseiten bzw. Anwendungen für mobile Geräte geschaffen. In den folgenden Kapiteln werden diese durch praktische Beispiele weiter vertieft. An guten Ansätzen und Ideen zur Umsetzung kann sich orientiert werden. Diese fließen in das Konzept der Anwendung BLoc mit ein.

In diesem Kapitel werden Anwendungen ausgewählt, untersucht und entsprechend bewertet. Für alle Anwendungen können die allgemeinen Richtlinien zur Usability aus der Norm DIN EN ISO 9241-110 sowie die Kriterien für mobile Webseiten verwendet werden. Für die nativen Apps sind zusätzlich noch die Empfehlungen für das jeweilige Betriebssystem zu beachten. Eine zusammenfassende Bewertung der Analyse mit den Noten 1 bis 4 ist in Anlage A zu finden. Die Bewertung mit einer 1 zeichnet eine sehr gute Umsetzung des jeweiligen Kriteriums aus, die Note 4 steht entsprechend für eine mangelhafte Umsetzung.

Die ausgewählten Anwendungen werden mit dem Betriebssystem iOS auf dem iPhone und dem iPad untersucht, da diese Geräte für Tests zur Verfügung stehen.

4.1 Auswahl der Anwendungen

Bei der Auswahl der Anwendungen wurde besonders auf die Ähnlichkeit und Vergleichbarkeit mit BLoc geachtet. Innerhalb von BLoc können die Auszubildenden ihr Berichtsheft täglich pflegen. Es besteht die Möglichkeit, größere Textmengen einzugeben und zu bearbeiten, sowie Dateien hinzuzufügen. Weiterhin sollen bestimmte Qualifikationen den einzelnen Tagen zugeordnet werden. Das Berichtsheft kann neben einer normalen Wochenansicht auch in einer Jahresübersicht angezeigt werden. In Abbildung 47 ist ein Ausschnitt des Berichtsheftes in BLoc zu sehen.

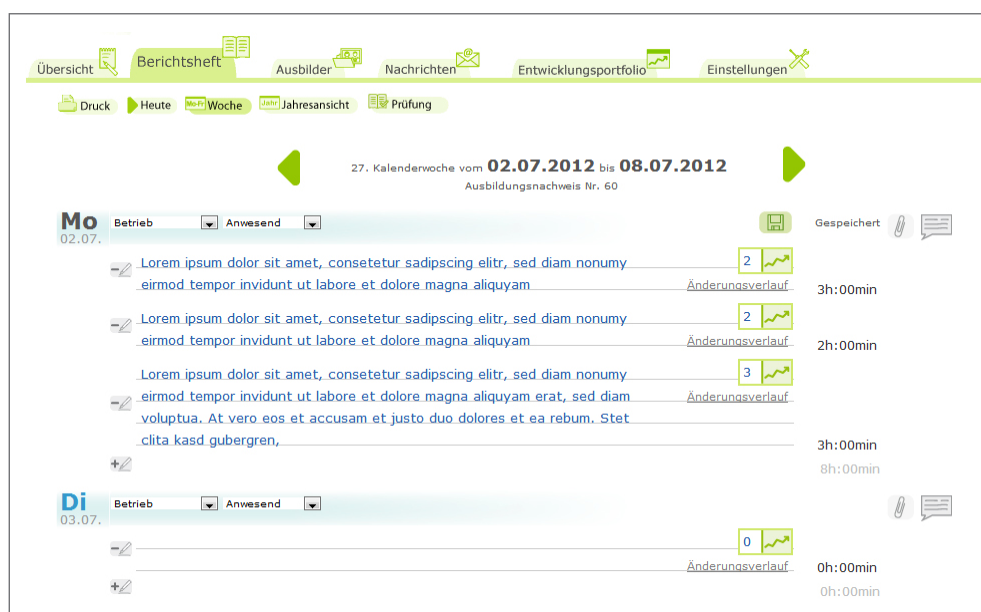


Abbildung 47 BLoc: Ansicht Berichtsheft

Für jeden Tag können verschiedene Einträge verfasst werden. Jedem Eintrag sind Qualifikationen entsprechend des Ausbildungsberufs hinzuzufügen und eine Zeit einzutragen. Mit den Auswahllisten, die für jeden Tag zur Verfügung stehen, können der Aufenthaltsort, zum Beispiel Betrieb sowie die Anwesenheit definiert werden.

Wurde eine Woche vollständig dokumentiert, kann sie den zuständigen Ausbildern freigegeben werden. Diese können nun die Berichtshefte einsehen und Kommentare und Hinweise für die Auszubildenden hinterlassen. Eine Woche kann akzeptiert oder abgelehnt werden. Nach der Ablehnung und Rückgabe an den Auszubildenden kann dieser die Woche erneut bearbeiten und verbessern.

Die Anwendungen für die Analyse sollten über erweiterte Möglichkeiten zur Text-Eingabe und Organisation von Einträgen verfügen, um eine Verbindung zu BLok herzustellen. Aus diesem Grund wurden ein digitales Notizbuch sowie ein E-Mail-Programm ausgewählt. Im Folgenden werden die Anwendungen kurz vorgestellt und anschließend bewertet.

4.2 Evernote - Online Notizbuch²⁷¹

Bei Evernote handelt es sich um eine Anwendung zur Erfassung und Verwaltung von Notizen. Es besteht die Möglichkeit verschiedene Notizbücher anzulegen, um eine Hierarchie zu schaffen und Inhalte zu sortieren. In diesen Notizbüchern können dann weitere Notizbücher und Notizen abgelegt werden.

Notizen können mit verschiedenen Metadaten versehen werden, zum Beispiel mit Schlagwörtern, einem Ort, dem Erstellungsdatum oder dem Datum der letzten Änderung. Weiterhin ist eine Funktion zum Teilen von Notizen über Facebook, Twitter, per E-Mail und per Link verfügbar. Ganze Notizbücher können außerdem freigegeben werden. Auf diesem Weg wird die Zusammenarbeit mit Kollegen oder Freunden möglich.

Evernote kann in Form einer Desktop-Anwendung für Windows und Mac OS X, als App zum Beispiel für die Systeme iOS und Android sowie als Web-Applikation über den Browser genutzt werden.²⁷² In Abbildung 48 ist ein Ausschnitt der Web-Applikation im Browser Google Chrome zu sehen. Auf der linken Seite ist die Übersicht über alle Notizbücher platziert. Daneben erfolgt die Auflistung der unterschiedlichen Notizen, die sich im aktuell ausgewählten Notizbuch befinden. Auf der rechten Seite wird die Notiz an sich und deren Metadaten angezeigt. Im Bearbeitungsmodus wird dem Benutzer ein einfacher Texteditor zur Verfügung gestellt.

271 Evernote Corporation : Evernote, Ihr virtuelles Gedächtnis

272 Vgl. Evernote Corporation: Evernote, Download

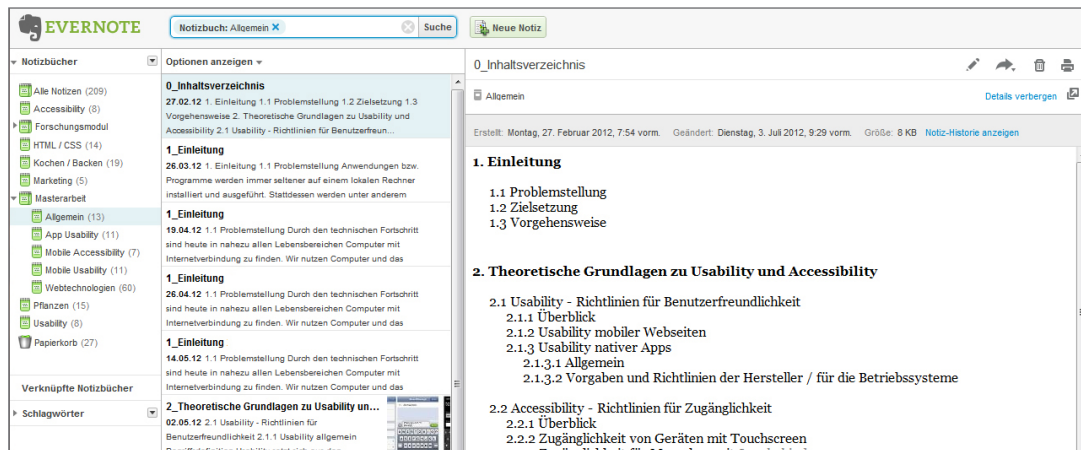


Abbildung 48 Evernote: Ausschnitt der Webanwendung

Die Webanwendung von Evernote ist nicht für mobile Geräte optimiert, d.h. es gibt keine mobile Webseite oder mobile Web-App. Aus diesem Grund wird nur die App für iOS auf dem iPhone und dem iPad untersucht.

4.2.1 Bewertung nach ISO 9241-110 Dialoggestaltung

4.2.1.1 Aufgabenangemessenheit

Die Anwendung Evernote ist sehr übersichtlich gestaltet und enthält nur die wichtigsten Funktionen. Dadurch wirkt die Oberfläche aufgeräumt und gut sortiert. Zwischen verschiedenen Ansichten kann schnell gewechselt werden, die nützliche Suchfunktion ist an vielen Stellen gut sichtbar platziert. Insgesamt muss der Benutzer keine unnötigen Dialogschritte durchführen und gelangt damit ohne Umwege zu den benötigten Funktionen.

Eine der wichtigsten Funktionen, das Hinzufügen einer neuen Notiz, auf die der Benutzer schnell Zugriff benötigt, ist jederzeit verfügbar. In Abbildung 49 ist zu sehen, dass diese beim iPhone stets zugänglich in der Mitte der Tab Bar platziert ist. Auf dem iPad ist „Notiz hinzufügen“ auf vielen Ansichten zu finden, da durch den größeren Bildschirm auch mehr Platz vorhanden ist. Die Funktion ist auf der Übersichtsseite der Notizbücher und auch bei der Ansicht eines speziellen Notizbuches, wie im rechten Bereich in Abbildung 49 ersichtlich, zu finden.

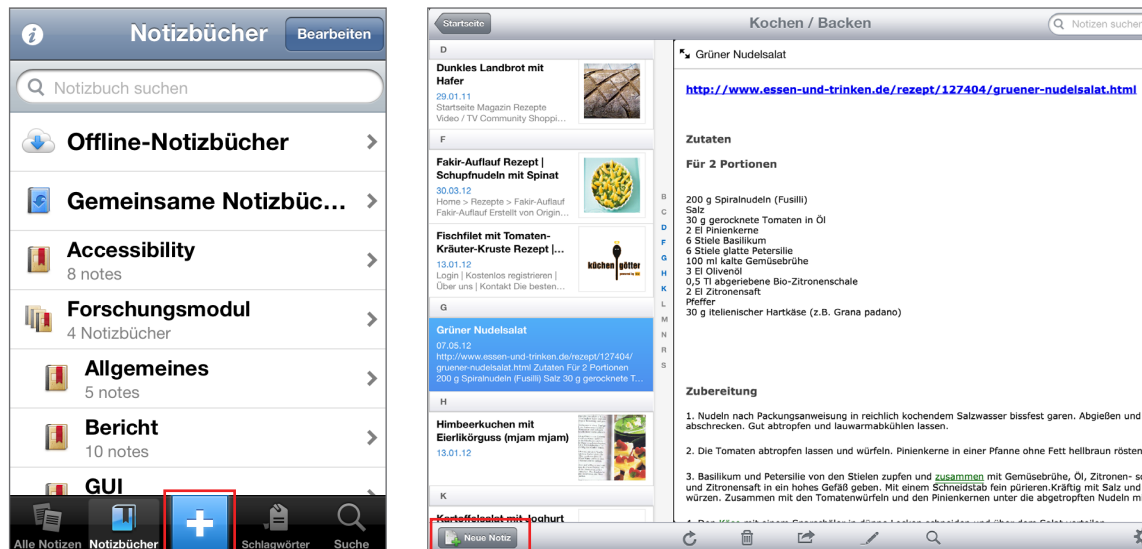


Abbildung 49 Evernote: Übersicht Notizbücher

Obwohl es sich bei Evernote um ein Notizbuch handelt, in das der Benutzer häufig Texte eingeben muss, werden diese auf das Wesentliche reduziert. An allen Stellen werden zahlreiche Vorgaben oder voreingestellte Werte zur Verfügung gestellt. Dies betrifft zum Beispiel das Hinzufügen von Schlagworten zu einer Notiz. Anstatt vom Benutzer zu verlangen, jedes Schlagwort von Hand einzugeben, wird ihm eine Liste mit bereits verwendeten Tags angeboten. Auf diese Weise kann der Nutzer sehr einfach und komfortabel Schlagwörter hinzufügen. Nur wenn ein neues Schlagwort vergeben werden soll, muss eine Texteingabe erfolgen. Auf diese Weise werden die notwendigen Eingaben auf die Notiz an sich beschränkt.

4.2.1.2 Selbstbeschreibungsfähigkeit

Für den Benutzer ist die Orientierung innerhalb einer Anwendung besonders wichtig und muss zu jeder Zeit gegeben sein. In Evernote wird diese Anforderung sehr gut umgesetzt. Jede Ansicht, die der Nutzer aufrufen kann, hat einen aussagekräftigen Titel. Somit wird dem Benutzer immer seine aktuelle Position angezeigt. Weiterhin ist in der Navigation Bar stets ein Zurück-Button mit der Aufschrift der vorherigen Ansicht verfügbar. Auch die Tab Bar, die in Abbildung 49 zu sehen ist, unterstützt den Benutzer bei der Orientierung durch die Anzeige der aktuellen Position. Nur die Ansicht einer einzelnen Notiz sollte noch verbessert werden. Dort wird der Titel der Notiz nicht in der Navigation Bar angezeigt, sondern im Inhalt. Ist der Text lang, muss der Benutzer scrollen, sodass der Titel nicht mehr sichtbar ist.

Auch die Übersicht über alle Notizen, zum Beispiel sortiert nach dem Titel, ist gut umgesetzt. In Abbildung 50 ist diese Variante auf dem iPad zu sehen. Es wird deutlich, dass jeder Buchstabe übersichtlich durch eine Überschrift abgetrennt ist. Außerdem wird dem Benutzer auf der rechten Seite die aktuelle Position bezogen auf den gesamten Inhalt angezeigt. In Abbildung 50 sind dementsprechend die Buchstaben R und S hervorgehoben.

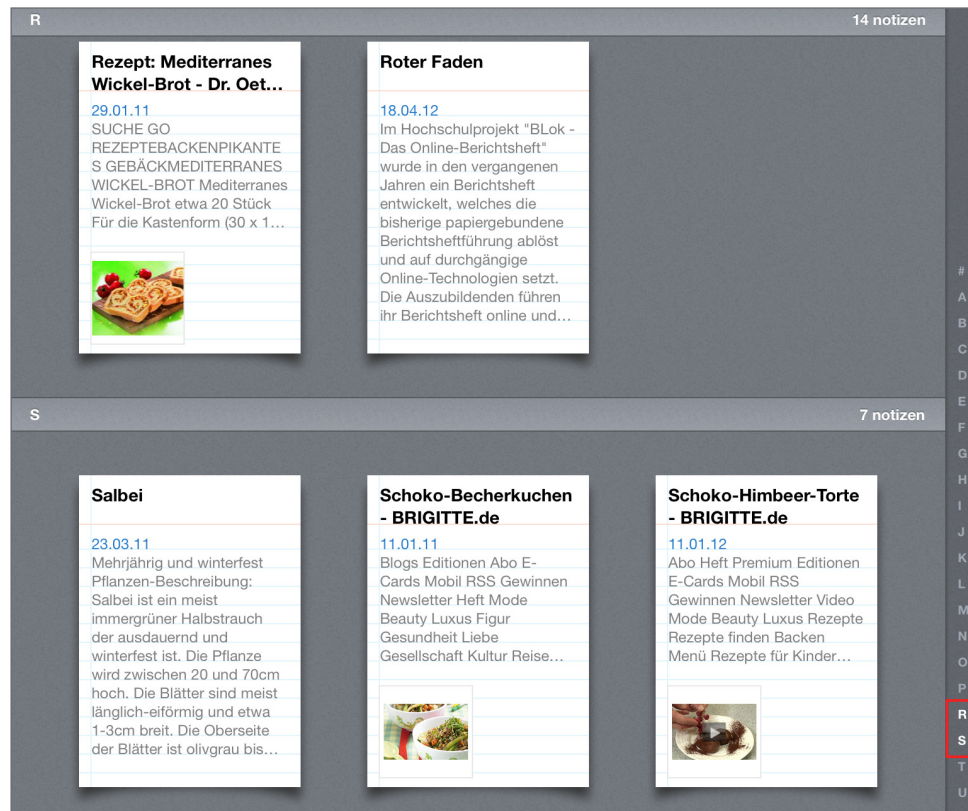


Abbildung 50 Evernote: Übersicht über Notizen auf dem iPad

Über erwartete Eingaben wird der Benutzer zum einen durch das Einblenden der Tastatur informiert. Weiterhin bietet Evernote den Hinweistext an, der in Abbildung 51 zu sehen ist. Durch diesen Hinweis wird der Benutzer nicht nur auf die notwendige Eingabe aufmerksam gemacht, sondern erhält außerdem eine Erklärung, wie er die Tastatur aktivieren kann.

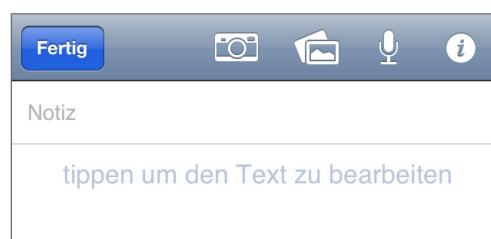


Abbildung 51 Evernote: Neue Notiz anlegen

In Abbildung 51 ist auch ersichtlich, dass an einigen Stellen Icons ohne Text für Funktionen genutzt werden. In den meisten Fällen ist das ausreichend, solange die Icons aussagekräftig genug sind, um dem Benutzer die Funktion deutlich zu machen. In Evernote werden zum großen Teil die Standard-Icons von iOS genutzt, zum Beispiel für die Funktionen Bearbeiten und Löschen. Die Benutzer sind mit diesen vertraut und können die zugehörige Funktion schnell zuordnen. Die benutzerdefinierten Symbole in Abbildung 51 sind ebenfalls eindeutig gestaltet und somit gut für den Benutzer interpretierbar. Mit dem Icon Kamera oder Mikrofon kann sehr schnell erfasst werden, dass der Benutzer an dieser Stelle Bilder oder Ton aufnehmen und der Notiz hinzufügen kann. Sollten die Funktionen doch unklar sein, kann der Benutzer diese sehr schnell durch das Aktivieren eines Icons erlernen. Hat er erkannt,

welche Funktion sich hinter dem jeweiligen Icon verbirgt, kann er sich diese durch die einfachen und bekannten Symbole schnell merken sowie sein Wissen wiederverwenden.

Die Selbstbeschreibungsfähigkeit der Anwendung Evernote ist größtenteils gut umgesetzt und kann positiv bewertet werden. Nur die Betitelung der Ansicht einzelner Notizen sollte wie vorgeschlagen verbessert werden, um die Orientierung des Nutzers überall zu ermöglichen.

4.2.1.3 Erwartungskonformität

Ein Kriterium innerhalb der Richtlinie Erwartungskonformität ist die Konsistenz, die in innere und äußere Konsistenz aufgeteilt wird. Die innere Konsistenz beschreibt das einheitliche Erscheinungsbild innerhalb der Anwendung. Bei der äußeren Konsistenz werden Betriebssystem und andere Anwendungen mit einbezogen.

Insgesamt ist Evernote innerhalb der Anwendung erwartungskonform gestaltet. Dem Benutzer wird es durch eine einheitliche Gestaltung von Buttons, Icons und Beschriftungen ermöglicht, gewonnene Erfahrungen anzuwenden. Auch die äußere Konsistenz, also die Konsistenz mit dem System iOS, ist gut umgesetzt. Eine detaillierte Bewertung der Konsistenz mit iOS findet sich in Kapitel 4.2.2.1.

Ein weiterer Bestandteil des Kriteriums Erwartungskonformität sind die Rückmeldungen für den Benutzer. Diese gibt es in Evernote in verschiedenen Formen. Zunächst ist die gesamte Anwendung sehr dynamisch gestaltet und so wie die meisten Apps animiert. Durch die Animationen werden dem Benutzer viele Aktionen verdeutlicht. Ist eine Notiz zum Beispiel noch nicht synchronisiert, erscheint über dieser ein Hinweis so wie im linken Bereich von Abbildung 52 zu sehen. Wurde die Synchronisation erfolgreich abgeschlossen, wird die Notiz zum aktuellen Monat verschoben. Durch das Animieren der Aktion „Verschieben“ wird dem Benutzer der erfolgreiche Abschluss des Vorgangs verdeutlicht. Ein weiterer Hinweis ist an dieser Stelle nicht mehr notwendig.

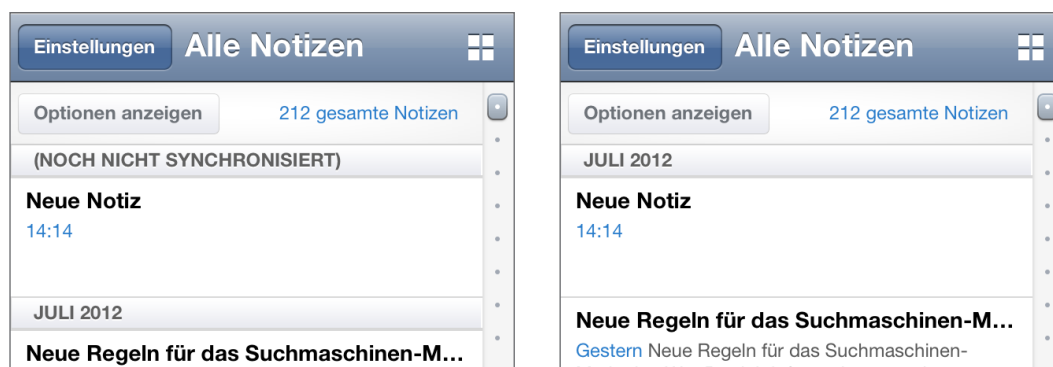


Abbildung 52 Evernote: Rückmeldung über Synchronisation einer Notiz

Wenn eine Aktion länger dauert als vom Benutzer erwartet, wird ebenfalls ein Hinweis angezeigt. Dies ist zum Beispiel beim Speichern von Notizen der Fall. Die Meldung für den Benutzer besteht aus einem animierten Symbol und dem Hinweis, welche Aktion gerade ausgeführt wird.

Insgesamt ist das Kriterium der Erwartungskonformität sehr gut umgesetzt. Besonders die innere Konsistenz ist positiv hervorzuheben. Eine einheitliche Gestaltung der Benutzeroberfläche sowie Positionierung der Elemente und die übersichtliche Struktur der Anwendung

unterstützen den Nutzer optimal bei der Verwendung von Evernote. Auch durch die Konsistenz mit dem System iOS fällt es dem Benutzer leicht, sich zu orientieren und seine Aufgaben zu erfüllen.

4.2.1.4 Lernförderlichkeit

Der Benutzer kann Evernote frei ausprobieren und die Verwendung der Anwendung somit leicht erlernen. Zu diesem Zweck stehen Rücksetzmöglichkeiten und Wiederherstellungsoptionen zur Verfügung. Zum Beispiel werden gelöschte Nachrichten in einem Papierkorb gesammelt und können von dort nach Bedarf wiederhergestellt werden. Bereits angelegte Notizen können außerdem stets bearbeitet und auch in andere Notizbücher verschoben werden. Der Benutzer ist demnach sehr flexibel bei der Arbeit mit Evernote. Notizen können nicht verloren gehen und unbeabsichtigte Fehler bei der Nutzung der Anwendung werden größtenteils vermieden.

Sind in Evernote noch keine Daten vorhanden, werden dem Benutzer in jeder Ansicht Erklärungen angeboten. Diese helfen ihm, die jeweilige Ansicht zu verstehen und diese mit Inhalten zu füllen. Die Hinweise verschwinden nicht, sondern bleiben stets erhalten, wie auch in Abbildung 53 dargestellt wird. Sie werden unter den angelegten Notizen angeordnet, sodass diese nicht von der Aufgabe ablenken oder Platz weg nehmen. Trotzdem kann der Benutzer jederzeit auf die Hinweise zurückgreifen und Details zur Verwendung der Funktionen nachlesen.

In Abbildung 53 sind zwei Beispiele der Hilfestellungen für den Benutzer zu sehen. Im linken Bereich der Abbildung wird beschrieben, wie Suchanfragen gespeichert werden können. Der rechte Bereich zeigt Erläuterungen zu gemeinsamen Notizbüchern. Auf diese Weise wird der Benutzer beim Verstehen und Verwenden der Anwendung Evernote unterstützt und kann die wichtigen Funktionen schnell erlernen.

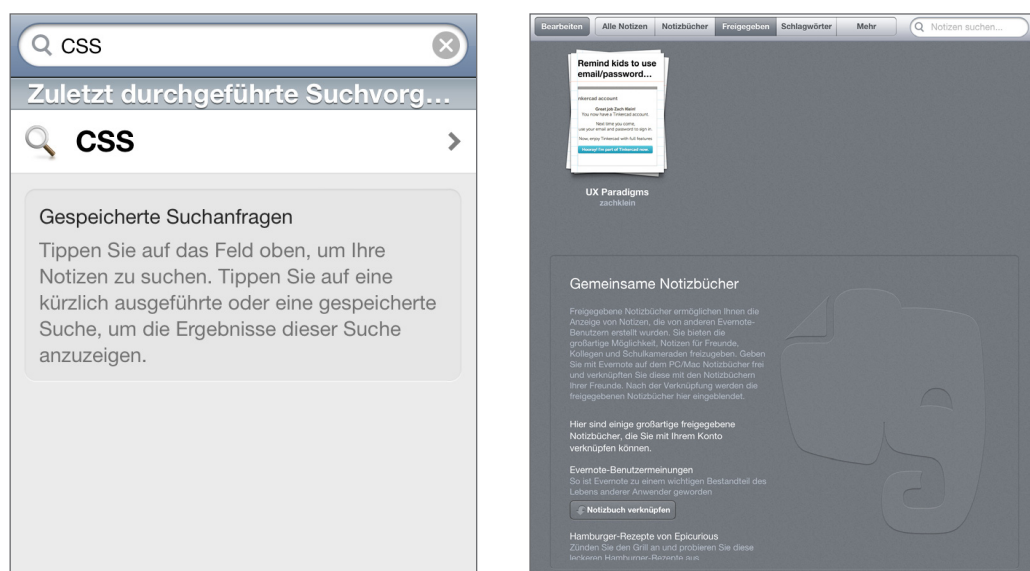


Abbildung 53 Evernote: Hinweise auf dem iPhone und iPad

4.2.1.5 Steuerbarkeit

Beim Kriterium Steuerbarkeit kommt es vor allem darauf an, dass der Benutzer die Kontrolle über die Anwendung hat und somit die Steuerung übernehmen kann. Bei Evernote wird diese Anforderung erfüllt, sodass der Benutzer die Arbeitsgeschwindigkeit stets selbst bestimmt. Der Benutzer hat weiterhin die Wahl, in welcher Reihenfolge er seine Aufgaben erfüllen möchte. Die Anwendung gibt ihm an dieser Stelle nichts vor, sodass eine flexible, individuelle Nutzung möglich ist.

Die Steuerbarkeit einer Anwendung schließt ebenfalls die Möglichkeit der Nutzung aller Ein- und Ausgabegeräte ein. Diese Anforderung kann im mobilen Kontext nur eingeschränkt betrachtet werden. Beim iPhone 4 stehen zum Beispiel nur der Touchscreen und VoiceOver als Screenreader zur Verfügung. Ab der Version 4S kann auch die Sprachsteuerung Siri genutzt werden. Eine kurze Übersicht zur Verwendung von Evernote mit VoiceOver wird in Kapitel 4.2.3 gegeben.

4.2.1.6 Fehlertoleranz

Die Fehlertoleranz ist eine der wichtigsten Richtlinien zum Thema Usability. Fehler können den Benutzer stark verunsichern und führen im schlimmsten Fall dazu, dass die Anwendung nicht mehr verwendet wird.

Grundsätzlich ist es wichtig, das Entstehen von Fehlern zu vermeiden. Evernote ermöglicht dies durch eine einfache Bedienung, gut erkennbare Symbole und die Wiederherstellungsmöglichkeit von gelöschten Notizen. Somit kann der Benutzer kaum unbeabsichtigt Fehler erzeugen. Dies gibt ihm Sicherheit bei der Arbeit mit der Anwendung, sodass die Nutzung von Evernote als positiv empfunden wird.

Da sich Fehler allerdings nie ganz vermeiden lassen, müssen verständliche und konstruktive Rückmeldungen mit einem Lösungsvorschlag angeboten werden. Bei Evernote kann es zum Beispiel ein Problem mit dem Synchronisieren der Notizen geben, wenn keine Internetverbindung besteht. An dieser Stelle sollte der Benutzer eine Rückmeldung bekommen. Dies ist bei Evernote aber nicht der Fall. Die Synchronisation wird schlicht nicht ausgeführt. Bei falschen Daten bei der Anmeldung wird dem Benutzer jedoch ein Hinweis angezeigt.

Während der Nutzung von Evernote sind keine Fehler oder Probleme aufgetreten, sodass die Anwendung positiv bewertet werden kann. Kleine Verbesserungen, wie zum Beispiel das Anbieten eines Hinweises bei fehlschlagender Synchronisation, können die Arbeit mit Evernote weiter optimieren.

4.2.1.7 Individualisierbarkeit

Die Richtlinie Individualisierbarkeit soll dem Benutzer Anpassungen des Systems an seine Bedürfnisse ermöglichen. Allerdings ist zu beachten, dass nicht alle Einstellungen des Nutzers auch sinnvoll oder hilfreich sind. Daher ist die Anpassbarkeit des Systems entsprechend zu beschränken.

In Evernote kann der Benutzer an einigen Stellen bestimmen, wie Informationen dargestellt werden. Zum Beispiel können Notizen nach Datum, Titel, Notizbuch und Ort sortiert werden. Durch die Einschränkung der Optionen kann der Benutzer nach seinen Bedürfnissen sortieren, ohne dabei die Benutzbarkeit versehentlich zu vermindern.

Ansonsten kann der Benutzer keine weiteren Anpassungen an seine Bedürfnisse vornehmen. Eine Änderung der Schriftgröße wäre aber aufgrund einer besseren Lesbarkeit in Evernote durchaus sinnvoll und sollte für den Benutzer ermöglicht werden.

4.2.2 Bewertung Usability für mobile Geräte

4.2.2.1 Konsistenz

Die mobile Webseite und die native App sollten stets konsistent mit der Desktop Webseite sein, um dem Benutzer ein Wiedererkennen zu ermöglichen. Bei Evernote ist das nur eingeschränkt der Fall. Die native App für iOS ist weder in dem für Evernote typischen Grün gehalten, noch werden sonstige Wiedererkennungsmerkmale wie Symbole verwendet. In Abbildung 54 wird dieses Problem verdeutlicht. Im linken Bereich ist die Webanwendung und rechts die native App unter iOS zu sehen. Gemeinsamkeiten der beiden Ansichten sind für den Benutzer zunächst eher nicht zu finden.

Dem Benutzer kann es daher unter Umständen schwer fallen, eine Verbindung zwischen Webanwendung und App herzustellen. Selbst das Logo von Evernote ist nicht oft in der App für iOS zu finden. Das Logo sollte im Kopfbereich der App stets sichtbar angeordnet sein. Weiterhin könnte die App mehr im für Evernote charakteristischen Grün gestaltet werden. Diese Verbesserung hilft dem Benutzer beim Herstellen des Zusammenhangs zwischen den beiden Anwendungen und bringt auch das Produkt insgesamt besser zur Geltung.

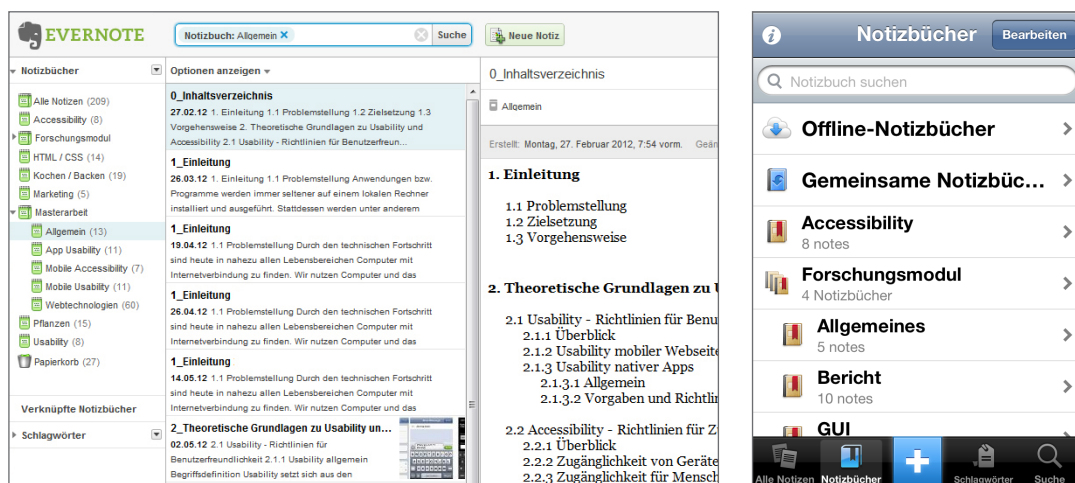


Abbildung 54 Evernote als Webanwendung und als native App unter iOS

Neben der Konsistenz zwischen den verschiedenen Arten der Anwendung sollten auch die Gestaltung und die Funktionen mit dem jeweiligen System konsistent sein. Die Anwendung Evernote orientiert sich sehr stark an den Vorgaben von Apple. Der Benutzer hat bei der Verwendung das Gefühl, dass es sich bei Evernote um eine fest integrierte App in iOS handelt. Dieser Umstand ist als sehr positiv zu werten, da Benutzer ihre Erfahrungen und Kenntnisse mit dem System sofort anwenden können.

Grundsätzlich sollte bei der Konsistenz ein Mittelweg gefunden werden. Eine native App sollte stets mit dem jeweiligen Betriebssystem konsistent sein. Diese Art der Konsistenz schließt aber nicht die individuelle Gestaltung des Produktes aus. Logo, Farben, Icons

und Schriftarten können entsprechend angepasst werden, ohne dass die Konsistenz zum Betriebssystem davon negativ beeinflusst wird.

4.2.2.2 Navigation

In Abbildung 55 ist zu sehen, dass Evernote auf dem iPhone eine am unteren Bildschirmrand fixierte Tab Bar für die Navigation zwischen verschiedenen Ansichten bietet. Das entspricht nicht nur den Empfehlungen von Apple, sondern ist für die Navigation auf kleinen Bildschirmen ideal. Die Tab Bar ist durch ihre Position schnell mit den Fingern erreichbar und der Benutzer verdeckt während der Verwendung keine wichtigen Elemente mit seiner Hand.



Abbildung 55 Evernote: Navigation zwischen den Ansichten auf dem iPhone und iPad

Auf dem iPad ist die Navigation anders positioniert. Im rechten Bereich von Abbildung 55 ist zu sehen, dass die Möglichkeit zum Wechseln zwischen den Ansichten oben in einer sogenannten „Segmented Control“ zu finden ist. Apple empfiehlt an dieser Stelle, das Element nur einzeln zu nutzen, selbst wenn genügend Platz zur Verfügung steht.²⁷³ In Evernote ist allerdings ein Bearbeiten-Button und die Such-Funktion neben der „Segmented Control“ in der Navigation Bar zu finden. Dadurch fehlt im oberen Bereich die Übersicht und der Benutzer könnte aufgrund der abweichenden Funktionen der drei Elemente eventuell verunsichert werden. Demnach sollten die anderen Funktionen mit im unteren Bereich untergebracht werden. Alternativ kann auch für das iPad die Tab Bar verwendet werden. Auf diese Weise wäre auch die Konsistenz zwischen iPhone und iPad App hergestellt.

Die Hierarchie der Navigation ist sehr flach und damit für den Benutzer übersichtlich und leicht nachvollziehbar. Maximal muss sich der Nutzer auf drei Ebenen bzw. Unterseiten bewegen. Zusätzlich wird immer ein Zurück-Button zur letzten Ansicht angeboten, der die Orientierung des Nutzers fördert.

4.2.2.3 Informationsdarstellung

Der Inhalt steht bei den meisten Webseiten und Anwendungen im Vordergrund. Daher sollte der wichtigste Inhalt direkt oben platziert sein und der Nutzer muss diesen auch ohne Scrollen sofort sehen können. Aufgrund des beschränkten Platzes bei mobilen Geräten müssen Kopfbereich und Navigation verkleinert werden. Evernote erfüllt diese Anforderung sehr gut, da beim iPhone eine am unteren Bildschirmrand fixierte Navigation verwendet wird. Weiterhin ist der Kopfbereich auf wenige Pixel beschränkt, sodass der Inhalt den meisten Platz auf

²⁷³ Apple: iOS Human Interface Guidelines, iOS UI Element Usage Guidelines

dem Bildschirm einnehmen kann. An einigen Stellen ist die Suche die wichtigste Funktion und somit im oberen Bereich noch vor dem Inhalt angeordnet.

In der App für das iPhone werden vorwiegend Listen genutzt, um viele Daten darzustellen. Diese Listen sind gruppiert und zum Beispiel nach Datum oder Titel sortiert. Auf diese Weise wirkt die App trotz einer großen Menge an Notizen sehr übersichtlich. Außerdem können die einzelnen Einträge sehr einfach mit den Fingern selektiert werden, was für eine fehlerfreie Navigation mit dem Touchscreen sorgt.

Innerhalb der iPad App ist die Übersicht über die Notizen aufgrund des verfügbaren Platzes anders gestaltet. Die Notizen werden als Dokumente dargestellt, die Titel und eine Vorschau bieten. In Abbildung 55 ist diese Art der Übersicht zu sehen. In diesem Fall wird die Metapher eines echten Notizbuches genutzt. Dem Benutzer fällt es aufgrund des Vergleiches mit der realen Welt sehr viel leichter, die Anwendung zu verstehen und zu benutzen.

4.2.2.4 Interaktionen

Für die Bedienung eines Touchscreens ist besonders die Größe und die Positionierung der Elemente wichtig. Die Größe sollte etwa der Auflagefläche eines Fingers entsprechen, also mindestens ein Zentimeter Höhe und Breite haben. Da die Anwendung Evernote ausschließlich Standardelemente des Systems iOS verwendet, ist die richtige Größe immer gegeben.

Weiterhin sollten die Elemente auch einen entsprechenden Abstand haben, damit der Benutzer nicht versehentlich den falschen Button auswählt. In Abbildung 56 ist ein Bereich von Evernote zu sehen, der diese Anforderung nicht erfüllt. Die Buttons „Einstellungen“ und „Optionen anzeigen“ liegen zu nah beieinander. Während des Tests wurde mehrmals versehentlich der falsche Button betätigt. An dieser Stelle sollte der Abstand noch etwas erhöht werden, um Fehler auszuschließen. Die sonstigen Interaktionselemente von Evernote erfüllen aber den Anspruch nach ausreichend Abstand.



Abbildung 56 Evernote: Platzierung der Buttons

Die wichtigen Elemente der Anwendung sollten so positioniert sein, dass der Benutzer bei deren Betätigung den Inhalt mit der Hand nicht verdeckt. Ideal ist daher eine Fixierung am unteren Bildschirmrand. Die meisten Funktionen in Evernote für das iPhone sind auf diese Weise angeordnet, sodass der Benutzer diese schnell erreichen kann und auch die Inhalte nicht verdeckt.

Innerhalb der App für das iPad sind die Funktionen auf den oberen und unteren Bereich verteilt. Die wichtigen Funktionen wie die Navigation und das Hinzufügen einer neuen Notiz

sollten unten positioniert werden, damit der Benutzer diese schnell erreichen kann und keine Inhalte verdeckt. Weniger häufig genutzte Funktionen können im oberen Bereich verbleiben.

4.2.2.5 Formulare

In Evernote sind die Texteingaben auf das Wesentliche reduziert. Der Benutzer muss nur den Titel und den Inhalt einer Notiz ausfüllen. Um sonstige Elemente, wie zum Beispiel die Schlagworte mit Inhalt zu füllen, werden stets Vorgaben angeboten. Der Benutzer kann aus diesen Vorgaben auswählen, wodurch sowohl die Eingabe an sich, als auch Eingabefehler vermieden werden.

In den leeren Eingabefeldern befindet sich ein Hinweis über die erwartete Eingabe für den Benutzer. Dies ist unter anderem beim Suchfeld oder dem Titel einer Notiz der Fall. Ein Beispiel ist im linken Bereich von Abbildung 57 zu sehen. Um dem Benutzer die Funktion des Eingabefeldes zu verdeutlichen, wird nicht nur ein Suchen-Symbol, sondern auch der Texthinweis „Notizbuch suchen“ verwendet.

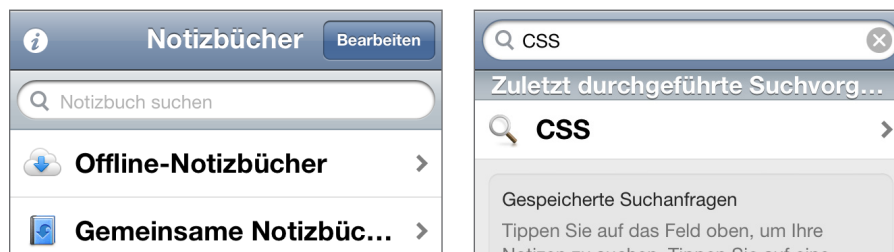


Abbildung 57 Evernote: Eingabefelder

Weiterhin wird für ein schnelles Löschen von Einträgen ein Icon direkt im Feld angezeigt, wie auch im rechten Bereich von Abbildung 57 zu sehen. Der Benutzer kann mit dieser Funktion den gesamten Eintrag auf einmal entfernen. Dadurch wird ihm das mühsame Entfernen jedes Buchstaben erspart und er kann stattdessen schneller mit seiner Aufgabe fortfahren.

4.2.3 Zugänglichkeit der Evernote App

Für den Test der Zugänglichkeit der Anwendung Evernote wurde der Screenreader VoiceOver unter iOS verwendet. Grundsätzlich kann der Benutzer Evernote sehr gut mit Hilfe des Screenreaders steuern, da zahlreiche Standardelemente von iOS eingebunden werden. Bei diesen ist die Zugänglichkeit bereits voreingestellt, sodass der Entwickler hier nur selten ergänzen muss. Dem Benutzer wird somit der Titel der Ansicht stets als Überschrift vorgelesen, der Zurück Button wird als solcher gekennzeichnet und Interaktionselemente werden zum Beispiel als „Button“ bzw. „Taste“ erkannt. In Abbildung 58 ist beispielhaft die Ausgabe von VoiceOver innerhalb einer Ansicht von Evernote dargestellt.

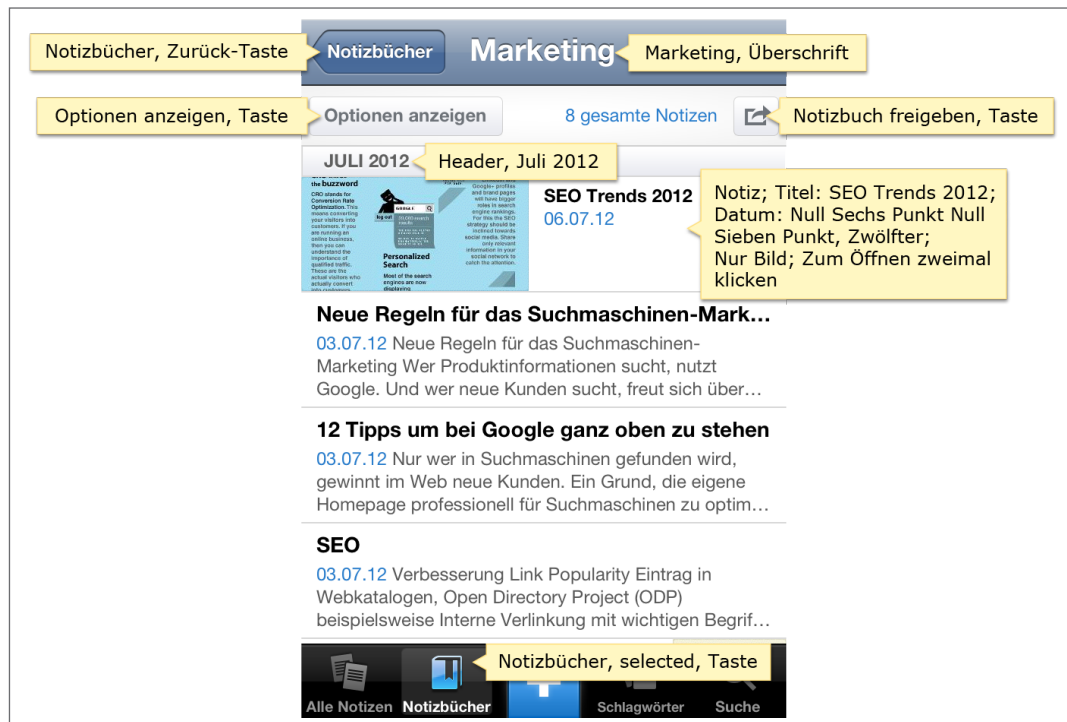


Abbildung 58 Evernote: Verwendung von VoiceOver unter iOS

Anhand von Abbildung 58 ist ersichtlich, dass der Benutzer zahlreiche nützliche Informationen über die Elemente auf dem Bildschirm bekommt. Es wird allerdings auch deutlich, dass es zum Beispiel beim Vorlesen des Erstellungsdatums einer Notiz Schwierigkeiten gibt. Der Benutzer wird zwar informiert, dass es sich um ein Datum handelt, es wird aber nicht entsprechend vorgelesen. Stattdessen hört der Benutzer folgenden Ausschnitt: „Datum: Null Sechs Punkt Null Sieben Punkt, Zwölfter“. Das Datum ist somit schwer verständlich und nicht mehr als solches erkennbar.

In der Liste aller Notizen ergibt sich für den Benutzer ein weiteres Problem. Jedes Listenelement enthält eine Notiz, welche durch das Auswählen geöffnet werden kann. Das jeweilige Listenelement enthält den Titel, das Datum und einen Ausschnitt des Inhaltes der Notiz. Durch VoiceOver werden diese Elemente vorgelesen. Allerdings nicht nur der Ausschnitt der zu sehen ist, sondern ein großer Teil des Inhaltes der Notiz. Der Benutzer muss aufgrund dessen sehr lange warten, bis der Hinweis „Zum Öffnen zweimal klicken“ ausgegeben wird. Besser wäre nur das Vorlesen des Titels der Notiz und des Datums. Der Inhalt sollte dem Nutzer erst nach dem Öffnen der Notiz vorgelesen werden.

Für den Benutzer kann es durch die falsche Sprache Probleme beim Verständnis geben. Einige Begriffe in Evernote, zum Beispiel „Notes“, „View Mode“ und „Arrow Up“ sind in Englisch verfasst, die Sprache für VoiceOver ist allerdings Deutsch. Aus diesem Grund werden die englischen Begriffe nicht entsprechend vorgelesen und sind somit zum Teil schwer verständlich. Wenn eine Anwendung in mehreren Sprachen angeboten wird, sollten auch alle Elemente möglichst mehrsprachig hinterlegt werden.

Sobald der Benutzer in eine andere Ansicht wechselt, wird immer zuerst die Überschrift vorgelesen, damit eine Orientierung möglich ist. Für Benutzer eines Screenreaders ist eine korrekte, sinnvolle Überschrift demnach noch viel wichtiger als für sehende Nutzer. In Evernote entsprechen die Überschriften zu einem großen Teil dieser Anforderung. Nur die

Überschrift der Detailansicht einer Notiz sollte verbessert werden. Ideal als Überschrift wäre der Titel der aktuell ausgewählten Notiz. In Evernote wird stattdessen aber die Nummer der Notiz im Vergleich zu den Gesamtnotizen angezeigt, zum Beispiel „1 von 8“. Dieser Zustand ist auch in Abbildung 59 zu sehen. Damit der Titel der Notiz vorgelesen wird, muss der Benutzer in die zweite Zeile navigieren und somit die wichtige Information erst suchen. Besser wäre das Anbieten des Titels der Notiz als Überschrift im Kopfbereich.



Abbildung 59 Evernote: Detailansicht einer Notiz

Die Texteingabe und Korrektur gestaltet sich für Nutzer von Screenreadern grundsätzlich schwierig. Der Benutzer muss zunächst den richtigen Buchstaben finden und auswählen. Anschließend wird der Buchstabe vorgelesen und der Benutzer kann diesen mit doppeltem Antippen schreiben. Danach muss der nächste Buchstabe gesucht werden. Zum Korrekturlesen des eingegebenen Textes gibt es verschiedene Optionen. Der Benutzer kann sich den gesamten Abschnitt oder einzelne Wörter vorlesen lassen. Auch eine Navigation von Zeichen zu Zeichen ist möglich. Besonders die letzte Methode ist für die Korrektur zum Beispiel eines Rechtschreibfehlers wichtig. In Evernote funktioniert das nur zum Teil. Bei der Navigation von Zeichen zu Zeichen wird zwar richtig vorgelesen, allerdings bewegt sich der Cursor nicht mit. VoiceOver ist zum Beispiel am Ende des Wortes angekommen, der Cursor steht aber noch in der Mitte des Wortes. Wenn der Nutzer nun nach dem Wort weiter schreiben will, werden die Buchstaben in diesem Fall in der Mitte des Wortes eingefügt. Dies ist für Nutzer eines Screenreaders sehr frustrierend und eine effektive Arbeit ist somit nicht möglich. Da das Verfassen von Notizen eine der wichtigsten Aufgaben in Evernote ist, muss diese Funktion verbessert werden.

Zu optimieren ist außerdem die Ausgabe von Hinweisen und Benachrichtigungen für die Nutzer von VoiceOver. Meldungen wie zum Beispiel das erfolgreiche Speichern einer Notiz oder das Synchronisieren des Evernote-Kontos sind nur für sehende Benutzer zugänglich. Durch VoiceOver werden diese dynamischen Inhalte nicht vorgelesen und wichtige Informationen bleiben verborgen.

4.2.4 Zusammenfassung Evernote

Die Anwendung Evernote kann insgesamt als positiv bewertet werden. Die Konsistenz mit dem System iOS wird eingehalten, der Benutzer kann somit seine Erfahrungen und Kenntnisse sehr gut anwenden. Auf diese Weise kann schnell in die Anwendung Evernote eingestiegen werden, ein Erlernen von Funktionen ist größtenteils nicht notwendig.

Schwierigkeiten gibt es allerdings mit der Konsistenz zwischen der Webanwendung und der iOS App. Zwischen diesen Anwendungen fällt es dem Benutzer schwer einen Zusammenhang herzustellen. Selbst das Logo der Anwendung ist nur auf dem Startbildschirm zu sehen und nur selten innerhalb der Anwendung. Auch das Farbkonzept von Evernote sollte auf die

App übertragen werden um nicht nur die Verbindung deutlich zu machen, sondern auch um das Produkt besser darzustellen.

Durch die Verwendung der iOS Standardelemente ist die Zugänglichkeit der Anwendung Evernote gut umgesetzt. Der Nutzer kann zu einem großen Teil effektiv mit VoiceOver arbeiten. Wichtig sind allerdings Verbesserungen beim Verfassen und Bearbeiten von Notizen. Durch Probleme bei der Texteingabe und Korrektur wird der Benutzer an dieser Stelle schnell frustriert und bei der Arbeit behindert.

4.3 Google Mail

Google stellt neben weiteren Anwendungen auch ein E-Mail Programm zur Verfügung. Die Benutzer können sowohl über den Browser als auch über die native App auf die Anwendung zugreifen. Die native App gibt es nur für iOS sowie für Android, wo diese bereits standardmäßig installiert ist.²⁷⁴ Für die Analyse kann demnach sowohl die native App unter iOS als auch die Web-App verwendet werden. Da sich beide Anwendungen grundsätzlich im Aussehen sehr ähnlich sind, werden diese gemeinsam analysiert. Der einzige auffällige Unterschied ist zunächst die Farbe. Die Web-App ist in einem hellen Blau gehalten, die iOS App dagegen in einem dunklen Grauton. Auf andere Unterschiede wird speziell an der entsprechenden Stelle hingewiesen.

Mit der Anwendung Google Mail können E-Mails abgerufen, sortiert und gesendet werden. Besonders die Navigation durch die verschiedenen Speicherorte und die Eingabe der Nachrichten ist für die Analyse interessant.

4.3.1 Bewertung nach ISO 9241-110 Dialoggestaltung

4.3.1.1 Aufgabenangemessenheit

Sowohl die Google Mail Web-App als auch die native App sind übersichtlich gestaltet und einfach gehalten. Die Hauptaufgaben der Anwendung sind E-Mails lesen und neue Nachrichten verfassen. Diese wichtigen Aufgaben können mit wenigen Schritten intuitiv ausgeführt werden. Weiterhin steht eine nützliche Suchfunktion direkt oben im Kopfbereich zur Verfügung. Weniger genutzte Optionen, wie zum Beispiel das Archivieren oder Löschen von E-Mails sowie das Verschieben in andere Ordner sind als Sammeloperation erst nach dem Markieren einzelner Nachrichten verfügbar. Im linken Bereich von Abbildung 60 sind die markierten E-Mails und die damit verfügbaren Optionen zu sehen. Diese Sammeloperationen ermöglichen ein schnelles und effektives Bearbeiten von mehreren E-Mails und erleichtern dem Nutzer somit die Arbeit.

274 Google Mobile: Google Mail für Handys

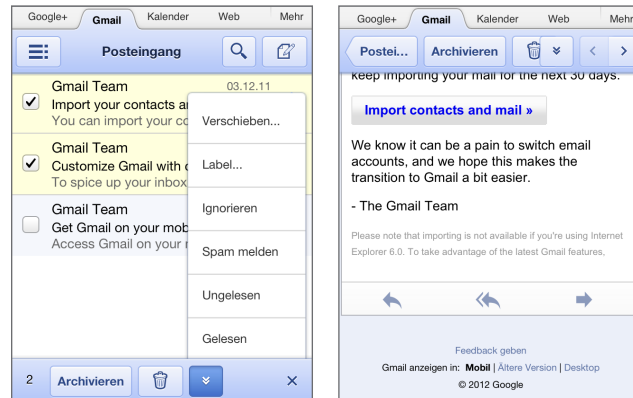


Abbildung 60 Google Mail: Optionen Posteingang und Detailansicht E-Mail

Weitere sehr wichtige Funktionen, wie das Antworten oder Weiterleiten von E-Mails sind im Rahmen der Detailansicht einer Nachricht verfügbar. Diese Ansicht ist im rechten Bereich der Abbildung 60 dargestellt. Bei Google Mail für das iPhone erreicht man die Funktionen aber nur durch Scrollen, sodass der Benutzer diese beim Aufruf der Nachricht zunächst nicht sieht. Da das Antworten bzw. Weiterleiten von Nachrichten an dieser Stelle wichtiger ist, als zum Beispiel das Archivieren, sollten die Funktionen präsenter gestaltet und stets sichtbar sein. Zum Beispiel könnten diese im Kopfbereich in der Navigation Bar zur Verfügung gestellt werden. Auf dem iPad sind die Funktionen aufgrund des größeren Bildschirms ohne Scrollen zu erreichen.

4.3.1.2 Selbstbeschreibungsfähigkeit

Die Anwendung Google Mail verfügt über eine übersichtliche und einfache Navigation. Es gibt maximal zwei Ebenen, sodass dem Benutzer die Orientierung leicht fällt. Weiterhin wird ein Titel oder ein Zurück-Button mit der Bezeichnung der vorherigen Ansicht angeboten. Dadurch ist es dem Benutzer möglich, seine aktuelle Position jederzeit zu erkennen.

Das Menü von Google Mail ist in Abbildung 61 zu sehen. Es wird entweder durch den Button oder durch ein Wischen mit einem Finger ein- und ausgeblendet. Der aktuelle Inhalt bleibt für den Benutzer aber immer noch sichtbar. Beim iPad ist dieser sichtbare Bereich, wie rechts zu sehen, größer da mehr Platz zur Verfügung steht.

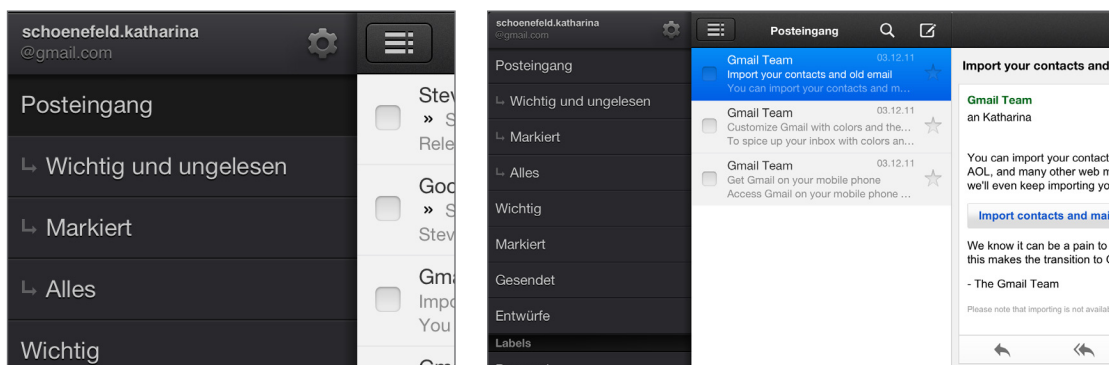


Abbildung 61 Google Mail: Menü auf dem iPhone und iPad

Beim iPad wird ein weiterer Vorteil deutlich. Zwischen der Übersicht über alle Nachrichten in einem Ordner und der Detailansicht muss nicht die Ansicht gewechselt werden. Die Übersicht bleibt auf der linken Seite für den Benutzer immer sichtbar, während im rechten Bereich die aktuell ausgewählte E-Mail angezeigt wird. Dies macht es dem Benutzer nochmals einfacher, sich innerhalb der Anwendung zu orientieren.

Text-Eingaben müssen vom Benutzer nur beim Verfassen oder Beantworten bzw. Weiterleiten einer E-Mail getätigt werden. Ruft der Benutzer ein solches Formular auf, wird der Fokus in das erste Eingabefeld gesetzt und die Tastatur aufgerufen. Auf diese Weise ist es für den Benutzer nicht nur eindeutig, dass eine Eingabe erwartet wird, sondern er kann diese auch einfach und intuitiv vornehmen.

4.3.1.3 Erwartungskonformität

Benutzer erwarten bei der Arbeit mit einer Anwendung Rückmeldungen und Feedback auf ausgeführte Aktionen. In Google Mail ist diese Anforderung sehr gut erfüllt. Wenn der Benutzer zum Beispiel Nachrichten löscht, wird ihm im unteren Bereich eine hervorgehobene Meldung angezeigt, dass der Vorgang erfolgreich war. Gleichzeitig besteht direkt die Möglichkeit, die Aktion rückgängig zu machen. Diese Ansicht ist in Abbildung 62 dargestellt.

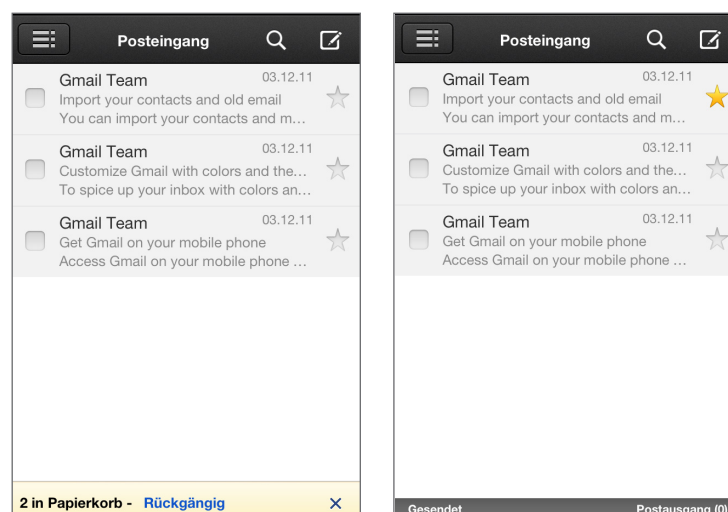


Abbildung 62 Google Mail: Rückmeldungen und Hinweise

Im rechten Bereich der Abbildung ist die Erfolgsmeldung nach dem Senden einer Nachricht zu sehen. Der Benutzer wird an dieser Stelle unaufdringlich auf die erfolgreiche Versendung hingewiesen. Da die Aktion abgeschlossen ist, verschwindet die Meldung nach einigen Sekunden wieder. Im Vergleich zur Meldung links in der Abbildung ist diese nicht farblich hervorgehoben, da es sich um einen neutralen, nicht kritischen Hinweis handelt.

Auch das Kriterium der inneren Konsistenz wird jeweils von der Web-App als auch von der nativen App erfüllt. Das Farbschema, die Gestaltung der Icons und die Positionierung der Elemente wird konsistent innerhalb der Anwendung verwendet. Somit kann sich der Benutzer schnell zurechtfinden und die Nutzung ist für ihn intuitiv. Details zur äußeren Konsistenz werden im nächsten Kapitel besprochen.

4.3.1.4 Lernförderlichkeit

Um eine Anwendung leichter erlernen zu können, ist es für den Benutzer immer von Vorteil wenn er Dinge ausprobieren kann. Damit eventuell ungewollte Aktionen rückgängig gemacht werden können, sollte eine Wiederherstellungsoption angeboten werden. In Abbildung 62 ist eine solche Möglichkeit im linken Bereich dargestellt. Dem Benutzer wird angeboten, die gerade gelöschten E-Mails wiederherzustellen. Auch später kann er diese Funktion noch nutzen, denn die Nachrichten werden zunächst nur in den Papierkorb verschoben.

Bei Google Mail gibt es keine kontextsensitive Hilfe. Dem Benutzer werden lediglich am Anfang, direkt nachdem er sich für den Dienst registriert hat, einige Information angeboten. Diese betreffen aber eher die gesamte Nutzung von Google Mail und nicht die Verwendung der Web-App oder der nativen App. Grundsätzlich ist Google Mail aber sehr einfach gehalten und viele Konzepte kennen die Benutzer bereits von E-Mail Programmen. Trotzdem gibt es einige Funktionen, wie zum Beispiel das Archivieren von Nachrichten, die anfangs unklar sind. An dieser Stelle sollten dem Benutzer Hinweise angeboten werden.

4.3.1.5 Steuerbarkeit

Die Anwendung Google Mail überlässt alle Aktionen dem Benutzer. Dieser kann die Geschwindigkeit und die Fortsetzung von Dialogen entsprechend seinen Bedürfnissen selbst bestimmen. Lediglich neue E-Mails werden automatisch heruntergeladen und angezeigt. Diese Funktion ist für eine solche Anwendung üblich und entspricht den Erwartungen der Benutzer.

Die Nutzung von allen Eingabe- und Ausgabegeräten ist im mobilen Kontext auf den Touchscreen und den Screenreader VoiceOver unter iOS beschränkt. Eine Zusammenfassung zur Verwendung von VoiceOver in Google Mail ist in Kapitel 4.3.3 zu finden.

4.3.1.6 Fehlertoleranz

Grundsätzlich sollten Fehler immer vermieden werden. Da dies aber nicht in jeder Situation möglich ist, muss der Benutzer unterstützt werden, eventuell auftretende Fehler zu erkennen und schnell beheben zu können. Bei der Anwendung Google Mail ist das nur zum Teil realisiert.

Ein möglicher Fehler ist zum Beispiel die Eingabe einer ungültigen E-Mail Adresse. Schon während der Bearbeitung der E-Mail könnte der Benutzer auf diesen Fehler hingewiesen werden. Allerdings bekommt er erst eine Fehlermeldung, wenn er die E-Mail absenden möchte. Die Fehlermeldung ist trotzdem gut umgesetzt, denn sie ist hilfreich und gibt dem Benutzer den richtigen Hinweis um den Fehler korrigieren zu können. In Abbildung 63 ist die Meldung dargestellt.

Eine weitere Verbesserung wäre die Auswahl des fehlerhaften Eingabefeldes nach dem Schließen der Fehlermeldung. Der Fokus liegt dann bereits auf dem jeweiligen Eingabefeld und die Tastatur ist eingeblendet, sodass der Benutzer direkt mit der Korrektur beginnen kann. Bei Google Mail ist dies allerdings nicht umgesetzt. Der Benutzer muss das fehlerhafte Feld erst identifizieren und auswählen, bevor er den Fehler korrigieren kann.



Abbildung 63 Google Mail: Hinweismeldung bei fehlerhafter Eingabe

Insgesamt ist die Anwendung Google Mail einfach und übersichtlich gehalten, sodass selten Fehler entstehen. Schwerwiegende Aktionen, wie zum Beispiel das Löschen von E-Mails, können wieder rückgängig gemacht werden. Bei wichtigen Schritten wird der Benutzer mit Hinweisen unterstützt. So bekommt er beim Schließen einer gerade bearbeiteten E-Mail die Optionen Verwerfen oder Speichern angeboten. Auf diesem Weg gehen keine Nachrichten versehentlich verloren. Kleine Verbesserungen sind bei Eingabefehlern notwendig, sodass an diesen Stellen Fehler schnell erkannt und korrigiert werden können.

4.3.1.7 Individualisierbarkeit

Der Benutzer hat innerhalb von Google Mail die Möglichkeit, E-Mails in verschiedene Ordner zu sortieren und diese somit zu organisieren. Zu diesem Zweck können aber nur die bereits vorgegebenen Kategorien genutzt werden. Der Benutzer hat keine Möglichkeiten, individuelle Ordner hinzuzufügen.

Weiterhin kann der Nutzer eine Signatur festlegen, die genutzt wird, wenn er E-Mails von der mobilen Version der Anwendung versendet. Auch das Konfigurieren einer Abwesenheitsnotiz ist möglich.

Zahlreiche weitere Einstellungen zur Personalisierung des Kontos sind nur über die Desktop-Version von Google Mail verfügbar. An dieser Stelle können Design, Ordner, Filter, Weiterleitungen usw. konfiguriert werden. Aufgrund des beschränkten Platzes und des Anspruchs der Einfachheit der mobilen Anwendung ist eine Reduktion des Funktionsumfangs durchaus positiv zu werten. Allerdings sollten wichtige Einstellung trotzdem auch im mobilen Kontext verfügbar sein. Der Benutzer sollte auch von unterwegs zum Beispiel eine Weiterleitung einrichten oder Filter verwenden können.

4.3.2 Bewertung Usability für mobile Geräte

4.3.2.1 Konsistenz

Die mobile Web-App und die native App für iOS sind sich sehr ähnlich. Der einzige nennenswerte Unterschied besteht in der Farbe der Anwendungen, wie auch Abbildung 64 verdeutlicht. Die native App ist eher dunkel gehalten, wie auch die Webanwendung für den Desktop bzw. die Produkte von Google insgesamt. Der Benutzer kann das Layout von Google Mail aber auch individualisieren, sodass die Farbe nicht für die Konsistenz zwischen den verschiedenen Arten der Anwendung genutzt werden kann. Daher ist es positiv, dass die Anordnung der Elemente und die sonstige Darstellung zwischen Web-App und nativer App größtenteils gleich sind.

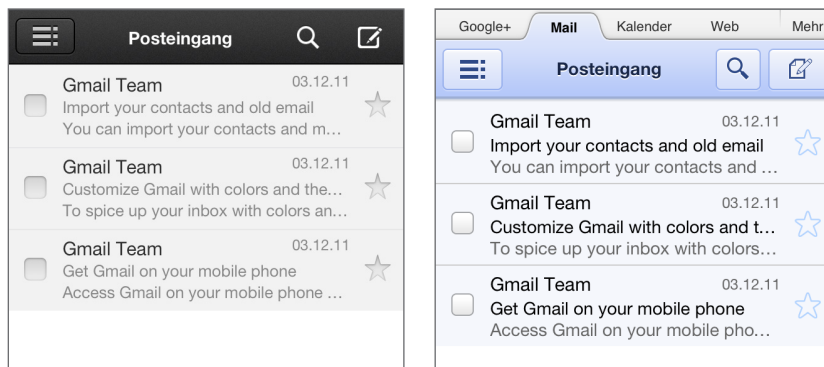


Abbildung 64 Google Mail: Native App für iOS und mobile Web-App im Vergleich

Eine native App sollte sich weiterhin stets am Aussehen und der Funktion des jeweiligen Betriebssystems orientieren. Bei Google Mail ist das zum Beispiel im Bezug auf die Navigation Bar und Toolbar nicht erfüllt. Für das Aufrufen von Optionen, die sich auf die aktuelle Ansicht beziehen, sollten Buttons am unteren Bildschirmrand bereit stehen, so wie im linken Bereich von Abbildung 65 bei der Mail-App von Apple zu sehen. Bei Google befinden sich diese für das Suchen und das Schreiben einer neuen Mail allerdings oben in der eigentlichen Navigation Bar. Dies ist auch rechts in Abbildung 65 zu sehen.

Durch diese Anordnung fehlt oben in der Navigation Bar auch der Platz für den vollständigen Titel der vorherigen Ansicht auf dem Zurück-Button. Weiterhin wird bei Google Mail keine Übersicht angezeigt, wie viele E-Mails es insgesamt im Ordner gibt und welche der Benutzer gerade liest. Bei der Mail-App von Apple ist diese Information im oberen Bereich platziert.

Im rechten Bereich von Abbildung 65 ist außerdem zu sehen, dass für einige Funktionen eine Drop Down Liste verwendet wird. Mit dem Pfeil-Button kann der Benutzer auf diese Weise weitere Optionen aufrufen, die für die aktuelle Ansicht verfügbar sind. Für das iPhone ist diese Funktion nicht üblich. Unter iOS sind solche Overlays bzw. Drop Down Menüs nur auf dem iPad verfügbar.

Kritisch ist weiterhin, dass wichtige Funktionen in diesem Menü versteckt sind. Zum Beispiel die Möglichkeit, E-Mails zu Verschieben oder zu Markieren. Diese Funktionen werden häufig genutzt, sind aber für den Benutzer auf den ersten Blick nicht zu entdecken. Da er auch nicht gewohnt ist, weitere Funktionen in einem versteckten Menü zu vermuten, wird er Schwierigkeiten haben, diese zu finden.

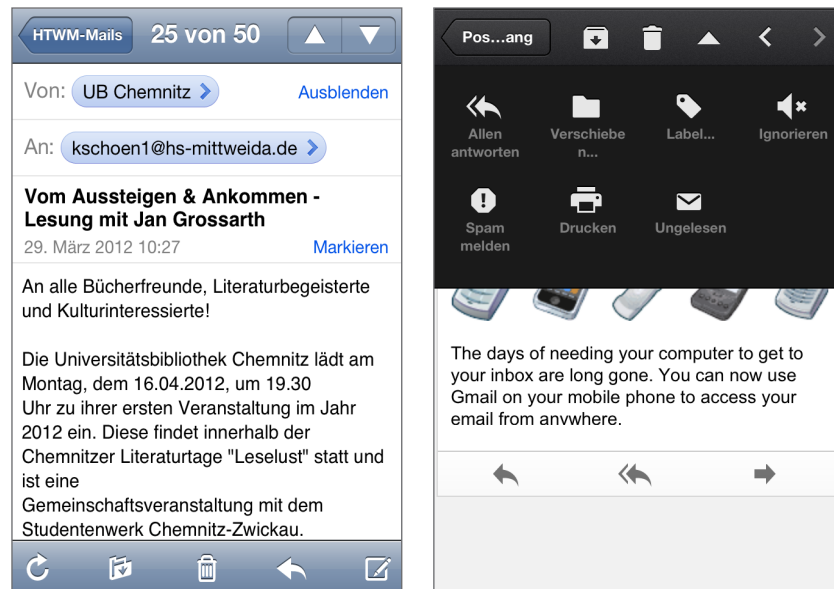


Abbildung 65 Mail-App von Apple und Google im Vergleich

Die Navigation zwischen unterschiedlichen Postfächern wird über einen Button in der Navigation Bar aufgerufen. Nach Betätigung des Buttons wird die Übersicht über die Postfächer von links ins Bild geschoben. Dieser Zustand ist in Abbildung 66 zu sehen. Die Navigation überdeckt den Inhaltsbereich aber nicht vollständig, sondern lässt im rechten Bereich diesen weiterhin sichtbar. Die Navigation kann auch durch das Wischen mit einem Finger ein- bzw. ausgeblendet werden.



Abbildung 66 Google Mail: Navigation zwischen einzelnen Postfächern

Diese Form der Navigation findet sich in keiner der von Apple standardmäßig installierten Apps wieder. Allerdings ist die Funktion immer häufiger zu finden, so zum Beispiel auch in der App von Facebook. Die Benutzer können an dieser Stelle von ihren eventuell vorhandenen Erfahrungen profitieren. Die Funktionsweise ist weiterhin sehr schnell zu erlernen, sodass an dieser Stelle für den Benutzer keine Nachteile entstehen, auch wenn diese Vorgehensweise nicht genau den Vorgaben von Apple entspricht.

4.3.2.2 Navigation

Die Navigation in Google Mail ist sehr einfach gehalten. Im vorherigen Kapitel wurde das Menü und dessen Funktionsweise bereits vorgestellt. Ansonsten muss der Benutzer nur zwischen der Übersicht der Nachrichten und der Detailansicht von E-Mails navigieren. Durch den mit der vorherigen Ansicht betitelten Zurück-Button ist das Wechseln zwischen den Ansichten sehr einfach. Dadurch muss der Benutzer nicht zu tief navigieren und behält jederzeit die Übersicht über seine aktuelle Position und die Möglichkeiten die er hat.

Ein weiterer Vorteil der Navigation ist die Platzersparnis. Auf diese Weise bleibt die Anwendung sehr übersichtlich und aufgeräumt. Trotzdem kann der Benutzer mit nur einer Geste das gesamte Menü aufrufen.

4.3.2.3 Informationsdarstellung

Der Fokus der Anwendung Google Mail liegt verstärkt auf dem Inhalt. Der Kopfbereich ist auf das Wesentliche reduziert, sodass ein Großteil des Platzes für den Inhalt bleibt. Weiterhin wird auch die volle Breite für die Listenansicht genutzt, in der die Nachrichten angeordnet sind. Dadurch sind die einzelnen Einträge leicht auswählbar, was die Bedienung der Anwendung mit dem Touchscreen unterstützt.

Im Kopfbereich der App sind zahlreiche Funktionen angeordnet und somit jederzeit sichtbar. Dies betrifft zum Beispiel die Suchen-Funktion sowie die Buttons zum Aufruf des Menüs und zum Senden einer neuen E-Mail. Daher sind die wichtigsten Funktionen an dieser Stelle für den Benutzer stets verfügbar. Anders ist es nur bei der Ansicht einer einzelnen E-Mail. Dort sind die wichtigen Funktionen zum Antworten oder Weiterleiten einer Nachricht unterhalb des Inhalts angeordnet und nur durch Scrollen erreichbar. Diese sollten in den oberen Bereich verschoben werden. Alternativ kann auch eine Toolbar genutzt werden. Dort können bis zu fünf Optionen bereitgestellt werden. Die Toolbar ist im Fußbereich angeordnet, aber trotzdem für den Nutzer immer sichtbar. Weiterhin wird in diesem Fall im Kopfbereich Platz für einen Titel der aktuellen Ansicht geschaffen.

4.3.2.4 Interaktionen

Um dem Benutzer zu verdeutlichen, dass er mit bestimmten Elementen interagieren kann, müssen diese entsprechend gestaltet sein. Der Großteil an Buttons und Icons erfüllt diese Anforderung. Besonders in der Web-App ist dies gut umgesetzt, wie auch im linken Bereich von Abbildung 67 zu sehen. Alle Interaktionselemente sind wie Buttons gestaltet, haben demnach eine Umrandung und einen hervorgehobenen Hintergrund.

In der nativen App sind nur einige Buttons als solche dargestellt, zum Beispiel der Zurück-Button im rechten Bereich von Abbildung 67. Die Icons daneben sind nicht auf diese Weise gestaltet. Dies ist problematisch, da das Vorgehen nicht den Erwartungen des Benutzers entspricht. Alle Funktionen, die in der Navigation Bar angeordnet sind, werden von Apple auch als Buttons gestaltet. Nur die Icons in der Toolbar können ohne Umrandung und Hintergrund dargestellt werden. In Google Mail sollten die Icons wie in der Web-App umrandet und mit einem anderen Hintergrund versehen werden. Alternativ können die Funktionen auch in eine Toolbar verschoben werden.

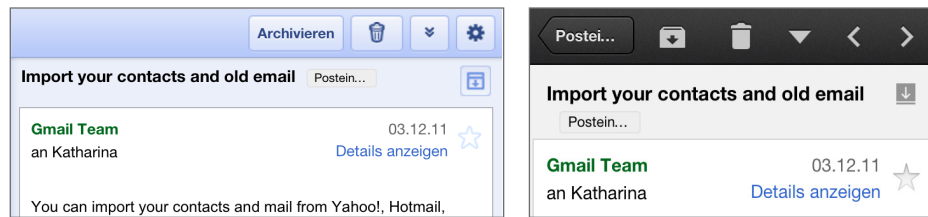


Abbildung 67 Google Mail: Gestaltung der Interaktionselemente

Neben der Gestaltung der Interaktionselemente ist auch deren Größe und der Abstand zueinander entscheidend. Die Größe der Icons und Buttons in Google Mail erfüllt die Anforderungen, der Abstand ist allerdings an einigen Stellen zu niedrig. Dies ist auch im rechten Teil der Abbildung 67 erkennbar. Die Icons liegen zu nah beieinander, sodass der Benutzer unter Umständen die falsche Funktion auswählt. Apple erlaubt zum Beispiel in der Toolbar nur fünf solcher Icons, diese nehmen dann aber auch die volle Breite des Bildschirms ein. Demnach sollte entweder diese Toolbar für die Funktionen genutzt werden oder die Optionen im Kopfbereich werden um mindestens eine reduziert.

4.3.2.5 Formulare

Die Formulare in Google Mail sind sehr einfach und übersichtlich gehalten. In Abbildung 68 ist die Ansicht zum Senden einer neuen Nachricht dargestellt. Eine kurze Bezeichnung der Eingabefelder hilft dem Benutzer beim Verständnis und ermöglicht ihm ein schnelles Ausfüllen.

Die Möglichkeit, eine komplette Zeile auf einmal zu löschen, sollte dem Benutzer auch angeboten werden. Unter iOS wird zu diesem Zweck ein Icon am Ende der jeweiligen Zeile eingeblendet.

Abbildung 68 Google Mail: E-Mail Formular

Die Eingaben sind bei Google Mail auf das Wesentliche reduziert. Der Benutzer muss nur Formulare ausfüllen, wenn er eine neue Nachricht schreibt oder wenn er auf eine bestehende antwortet bzw. diese weiterleitet.

4.3.3 Zugänglichkeit der Google Mail App

Der Test der Zugänglichkeit der nativen App sowie der Web-App Google Mail fand unter iOS mit dem Screenreader VoiceOver statt. Grundsätzlich gelten die in diesem Kapitel beschriebenen Verhaltensweisen sowohl für die native als auch für die Web-App. Auf Unterschiede wird zusätzlich hingewiesen.

In der App Google Mail werden kaum iOS Standardelemente verwendet, sodass es bei der Ausgabe durch den Screenreader Schwierigkeiten gibt. In Abbildung 69 sind beispielhaft einige Inhalte zu sehen, die dem Benutzer vorgelesen werden. Die Überschriften der einzelnen Ansichten werden zum Beispiel nicht als solche gekennzeichnet. Es wird wie in Abbildung 69 dargestellt, nur „Posteingang“ vorgelesen.

Weiterhin sind zahlreiche Elemente in englischer Sprache benannt. Dies betrifft vor allem die Buttons für das Menü, die Suche oder ähnliche Funktionen. Es kann nicht davon ausgegangen werden, dass der Benutzer die englische Sprache versteht. Weiterhin ist VoiceOver auf Deutsch eingestellt, sodass die englischen Begriffe nicht korrekt ausgesprochen werden und es dadurch zu zusätzlichen Verständnisproblemen kommen kann. In der Web-App sind dagegen alle Begriffe in der jeweiligen Sprache des Benutzers gehalten.

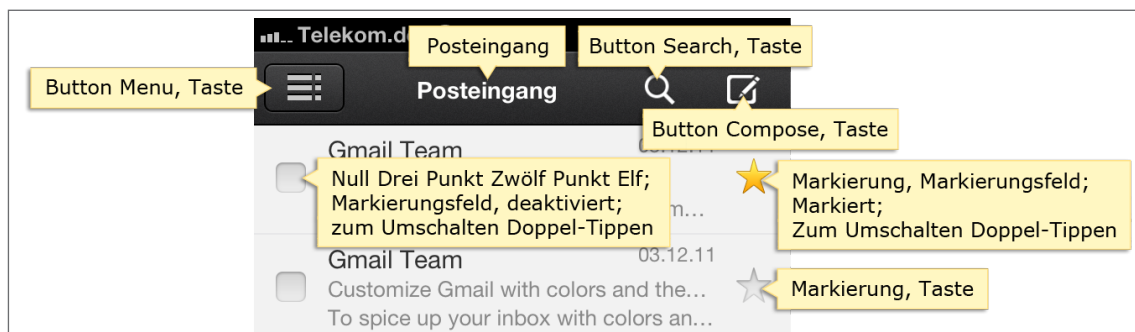


Abbildung 69 Google Mail: Verwendung von VoiceOver unter iOS - Ansicht 1

Besonders mit dem Menü, welches ebenfalls kein Standardelement unter iOS ist, haben die Nutzer von VoiceOver Probleme. Wird der Button für das Menü aktiviert, öffnet sich für die sehenden Benutzer die Navigation indem diese von links ins Bild geschoben wird. VoiceOver liest allerdings nur noch einmal den Button Menü vor. Die Nutzer können an dieser Stelle nicht merken, dass sich Inhalte verändert haben. Unter Umständen bleiben somit weitere Postfächer oder auch der Zugriff auf die Einstellungen verborgen. Selbst die Navigation von Element zu Element per Wischen nach Links oder Rechts funktioniert nicht korrekt. Ein Wischen nach Links ist nicht möglich, der Fokus bleibt auf dem Button Menü. Navigiert der Benutzer nach rechts, muss er erst den kompletten Inhalt durchgehen, bevor er das Menü erreicht. In diesem Fall springt der Fokus auch nicht an den Anfang des Menüs, sondern in die Mitte. In der Web-App von Google Mail ist das Menü besser umgesetzt. Dem Benutzer wird ausgegeben, dass sich Inhalte verändert haben. Er bekommt die E-Mail-Adresse des aktiven Kontos vorgelesen, da dies das erste Element im Menü ist. Besser wäre das Hinterlegen einer Überschrift, die beim Aufruf des Menüs immer ausgegeben wird. Weiterhin kann der Benutzer durch das Wischen nach rechts von Eintrag zu Eintrag navigieren. Die einzelnen Einträge werden auch als „Menü-Objekt“ gekennzeichnet.

Ein weiteres Problem ist die Orientierungsmöglichkeit der Benutzer. Den Wechsel zu einer anderen Ansicht können sehende Benutzer visuell wahrnehmen. Die neue Ansicht wird eingeblendet und enthält eine Überschrift um den Inhalt zu beschreiben. Den Benutzern von VoiceOver sollte ebenfalls der Wechsel zu einer neuen Ansicht und deren Inhalte ausgegeben werden. Bei Google Mail ist dies allerdings nicht der Fall. Anstatt der Überschrift wird beim Aufruf einer Ansicht ein beliebiges Element im Inhalt vorgelesen. Je nachdem wo sich der Button befindet, der zum Aufruf der Ansicht genutzt wird. Beim Wechsel von Inhalten sollte stets die Überschrift den Fokus erhalten und zuerst vorgelesen werden.

Auch der Aufruf des Kontextmenüs, welches in Abbildung 70 dargestellt ist, wird dem Benutzer nicht ausgegeben. Nach dem Markieren von E-Mails über die Checkbox sollte dem Benutzer die Inhaltsänderung mitgeteilt werden. Sollte der Benutzer das Menü am unteren Bildschirmrand trotz der fehlenden Kennzeichnung finden, ergeben sich weitere Probleme. In Abbildung 70 ist zu sehen, dass die einzelnen Funktionen nicht vorgelesen werden, sondern nur das Wort „Taste“. Ist der Benutzer auf VoiceOver angewiesen hat er keine Möglichkeit diese Funktionen zu erkennen und zu verwenden. Bei der Web-App bestehen die Funktionen dagegen aus Text und können dem Nutzer problemlos vorgelesen werden. Zusätzlich ist der Anfang und das Ende der Liste gekennzeichnet.

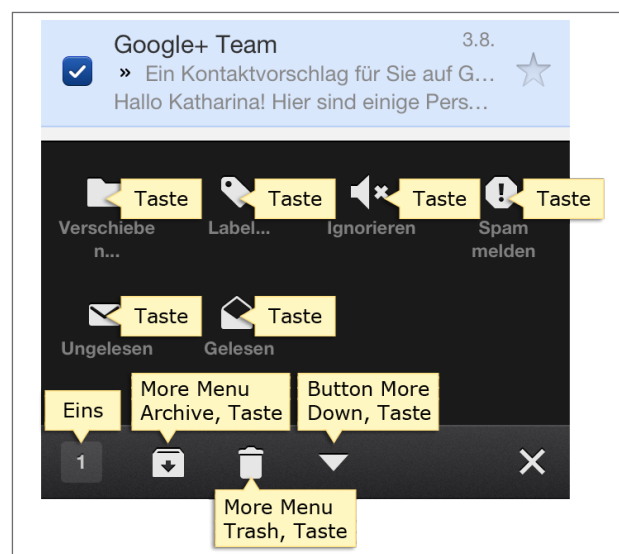


Abbildung 70 Google Mail: Verwendung von VoiceOver unter iOS - Ansicht 2

In der App Google Mail werden dynamische Hinweismeldungen genutzt um den Benutzer über den Status der Anwendung oder über erfolgreich ausgeführte Funktionen zu informieren. Diese Hinweise werden am unteren Bildschirmrand eingeblendet und nach einigen Sekunden wieder ausgeblendet. Gerade beim erfolgreichen Versenden, Löschen oder Verschieben einer Nachricht sollte die Meldung an den Benutzer durch VoiceOver weiter gegeben werden können. Wichtig wird dies vor allem, wenn die Funktionen nicht erfolgreich ausgeführt werden können und ein Fehler auftritt. Aktuell können diese dynamischen Inhalte nicht durch VoiceOver vorgelesen werden.

4.3.4 Zusammenfassung Google Mail

Sowohl die native App für iOS als auch die Web-App Google Mail sind sehr übersichtlich und einfach gehalten. Die wichtigsten Aufgaben, die der Benutzer erfüllen muss, stehen im Vordergrund, sodass ein effektives und effizientes Arbeiten möglich ist. Der Benutzer wird dabei nicht abgelenkt, aber trotzdem durch dezente Meldungen auf Wichtiges hingewiesen.

Die native App ist nur teilweise mit dem System iOS konsistent. Besonders fallen die Abweichungen die Navigation und das Kontext-Menü betreffend auf. Beide Elemente gibt es unter iOS nicht, sodass der Benutzer deren Funktionsweise erst erlernen muss. Für sehende Benutzer ist das Erlernen aber kein Problem, die Elemente können schon nach kurzer Zeit problemlos verwendet werden. Schwierigkeiten gibt es aber für Benutzer, die auf VoiceOver angewiesen sind. Die Zugänglichkeit dieser und auch anderer Elemente innerhalb der Anwendung Google Mail ist eher negativ zu bewerten. Dynamische Änderungen von Inhalten und das Aufrufen anderer Ansichten werden dem Benutzer nicht entsprechend kommuniziert. Benutzer von VoiceOver können ihr Aufgaben teilweise nur schwer mit Google Mail erfüllen. Besonders im Bereich Zugänglichkeit sind daher noch Verbesserungen notwendig.

4.4 Zusammenfassung der Analyse

Die Analyse ist als subjektiv zu werten, da diese nur durch einen Benutzer vorgenommen wurde. Ist eine objektive Betrachtungsweise notwendig, sollten mehrere Benutzer mit unterschiedlicher Erfahrung und verschiedenen Kenntnissen zu den Anwendungen befragt werden. Zu diesem Zweck kann ein Nutzertest durchgeführt werden. Bei diesem müssen die Benutzer mit den Anwendungen konkrete Aufgaben erfüllen. Dadurch zeigen sich weitere Probleme bei der Benutzung und beim Verständnis der Anwendungen.

Weiterhin sollte die Analyse von nativen Anwendungen nicht nur mit dem System iOS stattfinden, sondern zum Beispiel Android oder Windows Phone mit einbeziehen. Dadurch können Anregungen und Ideen zur Umsetzung von nativen Apps auch für weitere Systeme gewonnen werden. Im Rahmen der vorliegenden Arbeit war dies nicht möglich, da nur ein iPhone und ein iPad für die Tests zur Verfügung standen.

Trotz der genannten Einschränkungen bei der Analyse konnten wertvolle Erkenntnisse die Usability und die Zugänglichkeit betreffend gewonnen werden. Positive Beispiele zur Umsetzung von Funktionen oder der Darstellung von Inhalten können für die Gestaltung der Anwendung BLok abgeleitet werden.

5 Konzeption Benutzeroberfläche und Bedienung von BLok

Im Kapitel Konzeption wird die Erstellung des Konzeptes für die Anwendung BLok besprochen. Dieses Konzept enthält sowohl Vorgaben für die Benutzeroberfläche als auch für das Bedienkonzept von BLok. Neben einer allgemeinen Zusammenstellung von Richtlinien für die Gestaltung wird besonders auf die mobile Anwendung eingegangen.

5.1 Allgemeine Ansprüche an das Konzept

Die Anwendung BLok soll grundsätzlich einfach und übersichtlich gestaltet werden. Der Fokus liegt auf der Aufgabe des Benutzers und demnach hauptsächlich auf dem Führen bzw. Verwalten des Berichtshefts. Somit sollte die Gestaltung den Benutzer nicht ablenken oder irritieren. Die Reduktion auf die notwendigen Elemente und ein klarer Aufbau der Anwendung können diese Anforderungen erfüllen.

Zu beachten ist weiterhin, dass BLok von zwei unterschiedlichen Nutzergruppen verwendet wird. Zum einen sind dies die Auszubildenden, die ihr Berichtsheft pflegen und zum anderen die Ausbilder, die ihre Auszubildenden verwalten und deren Berichtshefte lesen und abnehmen. Für diese Nutzergruppen soll aufgrund der Wartbarkeit und Mandantenfähigkeit aber kein unterschiedliches Layout verwendet werden. Somit sind Konzeption und Gestaltung der Benutzeroberfläche unter Beachtung beider Benutzergruppen durchzuführen.

Für die Auszubildenden muss BLok frisch und ansprechend wirken und zudem zur Nutzung motivieren sowie Spaß an der Nutzung vermitteln. Für die Ausbilder sollte die Anwendung hingegen klar und aufgeräumt wirken sowie übersichtlich gestaltet sein, um ein schnelles Erfüllen der Aufgaben zu ermöglichen. Aus diesen Anforderungen ergeben sich zahlreiche Vorgaben an das Konzept der Anwendung. Zunächst muss die Struktur klar und nachvollziehbar sein und die Elemente sollten entsprechend der Arbeitsreihenfolge angeordnet werden. Besonders die komplexen Möglichkeiten im Rahmen des Berichtsheftes müssen gut verständlich und leicht erlernbar sein. Daher empfiehlt sich eine Einteilung in Bereiche und in eine Anordnung entsprechend der Wichtigkeit.

Damit sich die Auszubildenden besser mit BLok identifizieren und der Nutzung positiv gegenüber stehen, kann ein Profilbild hinterlegt werden. Auf diese Weise haben die Nutzer eine Individualisierungsmöglichkeit, durch die eine persönliche Verbindung zu BLok entsteht. Das Profilbild sollte entsprechend in allen Ansichten angezeigt werden und ist somit im Kopfbereich der Anwendung zu platzieren. Für eine Unterscheidung der Benutzerrollen wird eine farbliche Kennzeichnung verwendet.

Auch das Logo und der Name der Anwendung sollten für den Benutzer jederzeit sichtbar sein. Diese Elemente dienen der Identifizierung des Produktes und der Bekanntmachung der Marke. Weiterhin kann auf dem Logo ein schnell zu erreichender Link zur Startseite von BLok angeboten werden.

Neben dem Logo ist außerdem die Navigation im Kopfbereich der Anwendung zu platzieren. Um dem Benutzer schnell zu vermitteln, wobei es sich beim jeweiligen Navigationselement handelt, werden zusätzlich zum Text Icons verwendet. Diese müssen aussagekräftig und eindeutig sein, damit der Benutzer beim Verständnis unterstützt wird.

Trotz der zahlreichen Elemente im Kopfbereich der Anwendung muss dieser übersichtlich und einfach gestaltet werden. Besonders die Navigation muss im Vordergrund stehen. Der Benutzer nutzt diese Elemente nicht nur zum Navigieren, sondern auch zur Orientierung innerhalb der Anwendung. Daher sollte gerade das Menü selbsterklärend gestaltet und schnell erfassbar sein.

Für den Benutzer ist es weiterhin sehr wichtig, Funktionen zu erkennen und zu verstehen. Die Gestaltung und Positionierung der Elemente muss diesem Anspruch gerecht werden. Daher sollten Interaktionselemente stets entsprechend aussehen und in der Nähe des zugehörigen Inhaltes platziert werden. Funktionen sind demnach zum Beispiel als Buttons zu gestalten. Um dem Benutzer beim Verstehen der Funktion zu unterstützen können zusätzlich Icons verwendet werden. Links zu Seiten innerhalb und außerhalb der Anwendung sollten ebenfalls entsprechend dargestellt werden. Eine farbliche Differenzierung zum Inhalt und eine Unterstreichungen sind an dieser Stelle zum Beispiel erwartungskonforme Auszeichnungen.

5.2 Konzept für die Benutzeroberfläche

Bei der Erstellung des Konzeptes für die Benutzeroberfläche von BLok ist die Gestaltung der bisherigen Webanwendung zu berücksichtigen. Das Layout und der Aufbau der Anwendung, im Besonderen des Berichtsheftes, soll nicht zu deutlich verändert werden, da die Benutzer bereits mit der Oberfläche vertraut sind. Weiterhin wurde die Anwendung bereits im Bezug auf die Usability optimiert und auch durch Benutzer getestet, sodass viele positive Ansätze übernommen werden können.

Das Konzept für die mobile Anwendung muss sich ebenfalls an der Webanwendung orientieren. Ziel ist es, eine Anwendung für alle Geräte bereitzustellen und somit keine separate Anwendung für den mobilen Kontext anzubieten. Aus diesem Anspruch heraus entsteht die Abhängigkeit von der Struktur der Webanwendung sowie der Reihenfolge der Elemente.

Neben dem Konzept für die mobile Anwendung entsteht außerdem ein Entwurf für eine native App unter iOS. Die App kann zwar nicht komplett unabhängig von der Webanwendung sein, ist aber trotzdem flexibler. Grundsätzlich muss die Konsistenz zu BLok gewahrt bleiben, aber auch die Konsistenz zum System iOS.

Bei der Erstellung beider Konzepte muss weiterhin der mobile Kontext beachtet werden. Dieser umfasst nicht nur die Verwendung eines Touchscreens statt Maus und Tastatur, sondern auch eine wesentlich kleinere Fläche für die Inhalte. Zwischen der Größe von Smartphones und der Größe von Tablets muss unterschieden werden, sodass an dieser Stelle verschiedene Entwürfe entstehen. Außerdem ist auch der Wechsel der Orientierung des Gerätes zu berücksichtigen.

5.2.1 Konzept für die mobile Anwendung

Die bisherige Webanwendung BLok dient als Grundlage für das Konzept der überarbeiteten und der mobilen Anwendung. Die Webanwendung soll zu großen Teilen übernommen werden, da die Benutzer bereits mit dieser vertraut sind. An einigen Stellen innerhalb der Anwendung, zum Beispiel im Berichtsheft, werden kleinere Verbesserungen der Struktur und

das Layout betreffend durchgeführt. In Abbildung 71 ist ein Ausschnitt des Konzepts für das Berichtsheft zu sehen. Die vollständige Ansicht ist in Anlage B hinterlegt.

Abbildung 71 BLok: Entwurf für das Berichtsheft

Im Vordergrund bei der Gestaltung standen besonders Einfachheit und Übersichtlichkeit. Nur die notwendigen Elemente werden angezeigt und entsprechend der Arbeitsreihenfolge angeordnet. Außerdem wurde auf eine bessere Abtrennung in Bereiche geachtet, sodass die einzelnen Tage, aber auch die jeweiligen Einträge besser voneinander unterscheidbar sind. Dem Benutzer wird nun klarer, dass bestimmte Optionen für einen Tag und andere wiederum nur für einen Eintrag verfügbar sind.

Auch die Pflichtfelder sind nun besser von den optionalen Funktionen „Datei anhängen“ und „Kommentar verfassen“ sowie der Information, ob der Eintrag bereits gespeichert wurde, abgegrenzt. Somit kann sich der Benutzer auf die wichtigen Pflichtangaben konzentrieren ohne von optionalen Möglichkeiten abgelenkt zu werden.

In Abbildung 72 ist zum Vergleich ein Ausschnitt des aktuellen Berichtsheftes zu sehen.

Abbildung 72 BLok: Ansicht Berichtsheft

Weiterhin wurde die Navigation besser von der Möglichkeit, die Ansicht zu Wechseln getrennt. Bisher sind beide Arten von Funktionen in der Unternavigation zusammengefasst.

Im Entwurf wurde daher die Unternavigation auf „Berichtsheft bearbeiten“, „Berichtsheft drucken“ und „Prüfung“ beschränkt. Der Wechsel zum aktuellen Tag mit dem Button „Heute“ wird direkt neben der Funktion von Woche zu Woche zu blättern angeordnet. Die Möglichkeit von der Wochen- in die Jahresansicht zu wechseln bekommt einen eigenen Bereich. Durch die Gruppierung der Elemente entsprechend ihrer Funktion wirkt die Oberfläche aufgeräumt und dem Benutzer wird das Verständnis erleichtert.

Die Gestaltung der Interaktionselemente wird ebenfalls verbessert, um dem Benutzer das schnelle Erkennen dieser zu ermöglichen. Der Benutzer muss zum Beispiel bei jedem Eintrag eine Zeit hinterlegen. Das Eingabefeld ist aber nicht als solches dargestellt, sodass die Interaktion unter Umständen nicht klar wird. Im Berichtsheft sollten alle Interaktionen wie Eingabefelder oder Buttons entsprechend dargestellt werden, um dem Benutzer das Erkennen von Funktionalitäten zu erleichtern.

Von der Positionierung und Gestaltung der Elemente hängt nun das Konzept für die mobile Anwendung ab. BLok sollte für alle Bildschirmgrößen und demnach auch für die kleinen Bildschirme der mobilen Geräte geeignet sein. Da vordergründig nur mit Hilfe des Layouts eine Anpassung an die unterschiedlichen Bildschirmgrößen erreicht werden soll, muss sich zunächst an der Struktur der Webanwendung orientiert werden. Später sollten bestimmte Bereiche wie der Header und die Navigation für mobile Geräte ausgetauscht werden. In Abbildung 73 ist ein Ausschnitt des Entwurfes für die Startseite und das Berichtsheft für kleine Bildschirme zu sehen. Die vollständige Ansicht dieser beiden Entwürfe sowie der Entwurf für die Ansicht des Ausbilders sind in Anlage B zu finden. Die aufgrund der Höhe nicht mehr sichtbaren Bereiche sind mit einer transparenten grauen Fläche überdeckt. Der Benutzer muss demnach Scrollen, um die restlichen Inhalte zu erreichen.

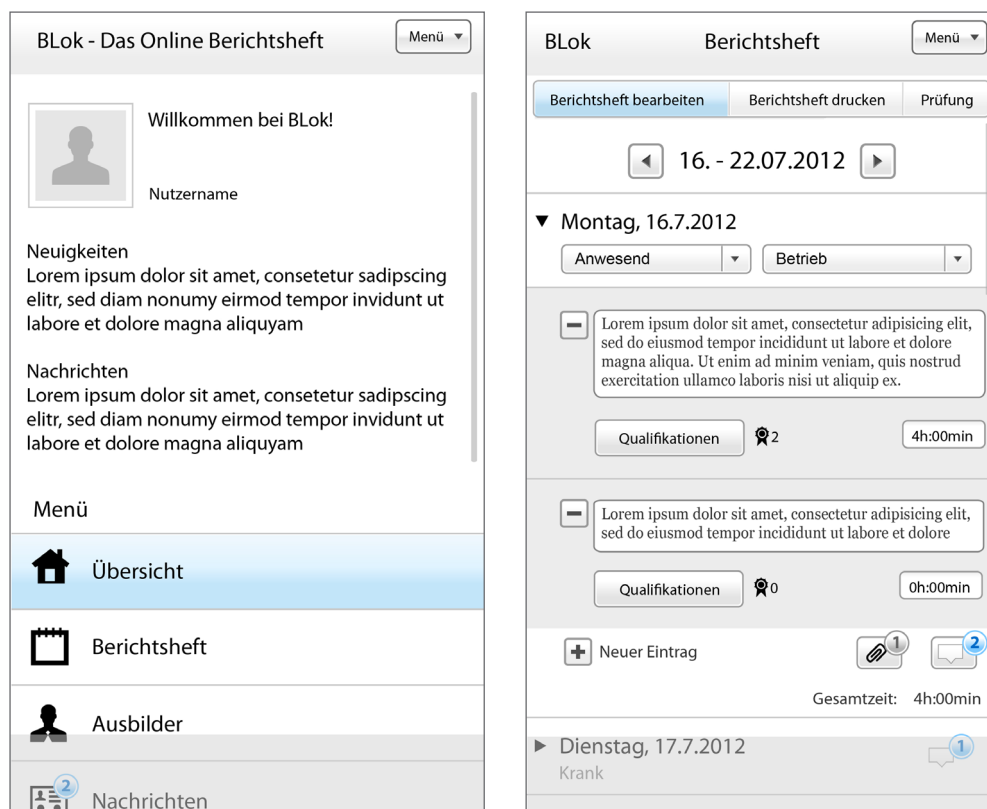


Abbildung 73 BLok: Entwurf mobile Webanwendung - Smartphone

Der Benutzer sollte trotz der beschränkten Größe der mobilen Geräte ohne Scrollen bereits einen Großteil des Inhaltes sehen. Daher wurde der Kopfbereich der mobilen Anwendung stark reduziert. Im Header sind nur noch der Titel der Anwendung bzw. die Überschrift der aktuellen Seite sowie der Button zum Aufrufen der Navigation zu sehen. Das Bild des Benutzers wurde ebenfalls aus dem Kopfbereich entfernt und ist nur noch auf der Startseite verfügbar. Durch die genannten Maßnahmen konnte der Kopfbereich so verkleinert werden, dass ein Großteil des Inhaltes bereits ohne Scrollen sichtbar und zugänglich ist.

Die Navigation befindet sich unter dem Inhalt und nimmt daher im oberen Bereich keinen wichtigen Platz weg. Über den Button „Menü“ im Header kann diese zu jeder Zeit aufgerufen werden. Ein weiterer Vorteil der neuen Positionierung ist der verfügbare Platz für die Navigation an sich. Die Menüeinträge können über die gesamte Breite des Bildschirms gehen und der Benutzer kann diese dadurch leicht mit den Fingern aktivieren. Außerdem bleibt Platz für Icons sowie optionale Hinweise, zum Beispiel bei einer neuen Nachricht.

In einigen Ansichten, unter anderem im Berichtsheft, wird außerdem eine Unternavigation benötigt. Diese ist direkt unter dem Kopfbereich angeordnet, damit der Benutzer schnell zwischen den verschiedenen Inhalten navigieren kann. Das Menü wird in Form von Buttons dargestellt, da aufgrund der Interaktion mit dem Finger mehr aktive Fläche benötigt wird. Dadurch wird sicher gestellt, dass der Benutzer das richtige Element bequem auswählen kann und nicht aus Versehen den nebenstehenden Menüpunkt aktiviert.

Im Berichtsheft wird dem Benutzer eine Liste mit den einzelnen Tagen als Übersicht angeboten. Wählt er einen Tag aus, wird dieser in der aktuellen Ansicht geöffnet und kann bearbeitet werden. Öffnet der Benutzer einen anderen Tag, wird der vorher ausgewählte wieder geschlossen, um die Seite übersichtlich zu halten. Die wichtigsten Angaben zum jeweiligen Tag werden bereits im geschlossenen Zustand angezeigt. Dabei handelt es sich um Informationen wie die Anwesenheit, den Ort, die gebuchte Zeit sowie eine Information über neue Kommentare. Die Übersicht bietet dem Benutzer auf diesem Weg alle wichtigen Informationen auf einen Blick an. Er muss nur in die Detailansicht wechseln, wenn er seine Einträge bearbeiten will.

Die Positionierung einiger Elemente, zum Beispiel des Menüs, weicht vom Konzept der Webanwendung ab. Diese neue Strukturierung ist aber wichtig für die Benutzbarkeit von BLok auf einem mobilen Endgerät mit kleinem Bildschirm. Daher müssen diese Elemente speziell für den mobilen Kontext angepasst und auch nur den Nutzern von mobilen Geräten zur Verfügung gestellt werden.

Für Tablets und Netbooks wurde ein dritter Entwurf entwickelt, der in Abbildung 74 zu sehen ist. Diese Geräte verfügen über größere Bildschirme und bieten somit mehr Platz für die Anwendung. Trotzdem kann an dieser Stelle nicht das Layout für die Desktop-Variante genutzt werden, da dieses zu breit angelegt ist.

Die Elemente im Kopfbereich werden für Tablets und Netbooks nicht neu positioniert, sondern nur in der Größe angepasst. Das Profilbild des Benutzers sowie die Navigation wurden zum Beispiel verkleinert. So kann die Bildschirmgröße von Tablets besser ausgenutzt werden, ohne dass der Benutzer Schwierigkeiten bei der Bedienung hat oder das Layout zerstört aussieht.

Im Entwurf des Berichtshefts für Tablets wurden die Elemente zum Hinzufügen von Dateien und Kommentaren unterhalb des Bereichs mit den Einträgen platziert. Auf diese Weise

können das Textfeld sowie der Button und das Eingabefeld für die Zeit die gesamte Breite einnehmen.

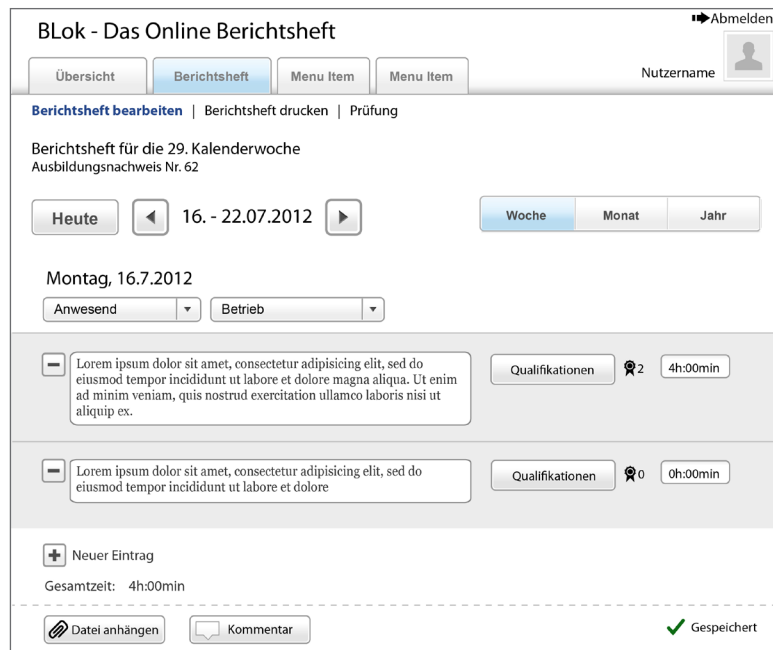


Abbildung 74 BLok: Entwurf mobile Webanwendung - Tablet

5.2.2 Konzept für die iOS App

Für BLok wird zunächst die Umsetzung einer mobilen Webanwendung angestrebt, da nicht genügend Ressourcen für die Entwicklung von nativen Apps zur Verfügung stehen. Weiterhin sollen Meinungen und Reaktionen der Benutzer abgewartet werden. Wird die mobile Anwendung häufig eingesetzt und stehen die Nutzer diesem Thema positiv gegenüber, kann erneut über das Anbieten von nativen Apps für zum Beispiel iOS oder Android nachgedacht werden. Trotz der geplanten Umsetzung der Webanwendung wurde beispielhaft ein grobes Konzept für die iOS App erarbeitet. Somit besteht bereits eine Grundlage, sollte sich doch für die Erstellung einer nativen App entschieden werden.

Grundsätzlich sollte bei der Erstellung einer nativen App stets auf die Konsistenz mit dem Betriebssystem geachtet werden. Dies hilft dem Benutzer beim Verstehen und Verwenden der Anwendung, da er seine Erfahrungen mit dem System auch auf die App übertragen kann. In Abbildung 75 sind die Entwürfe für die Startseite, die Übersicht über das Berichtsheft und die Ansicht eines einzelnen Tages zu sehen. Für einen großen Teil der Daten wird die Table View von iOS, eine Listenansicht, verwendet. Diese ermöglicht das Darstellen von Icons und Informationen den jeweiligen Eintrag betreffend. Weiterhin kann die Listenansicht zur Navigation innerhalb der Anwendung genutzt werden. Somit navigiert der Benutzer über einen Eintrag zum Berichtsheft und von dort aus zur Detailansicht eines Tages.

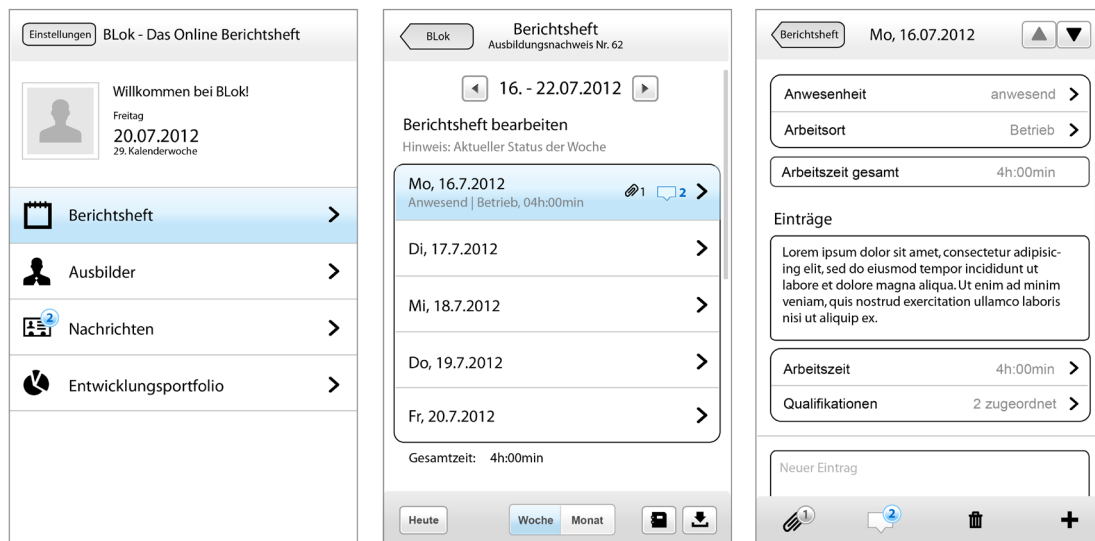


Abbildung 75 BLok: Entwurf iOS App

Zur Orientierung und zum Navigieren steht im Kopfbereich ein Zurück-Button mit der Aufschrift der letzten Ansicht zur Verfügung. Der Benutzer weiß somit stets, wo er sich befindet und kann an diese Stelle zurückkehren. Durch die Überschrift der aktuellen Ansicht direkt daneben wird die Orientierung weiter verbessert. Diese Elemente sind in der Navigation Bar von iOS untergebracht, sodass die Konsistenz mit dem System gewahrt ist.

Auch die Toolbar, die zum einen die Navigation zwischen den verschiedenen Ansichten des Berichtsheftes enthält, sowie die Funktionen in der Detailansicht, sind Standardelemente von iOS. Die Toolbar ist am unteren Bildschirmrand fixiert, sodass der Benutzer jederzeit auf die Funktionen zugreifen kann, auch wenn er im Inhalt scrollen muss. Die Elemente ermöglichen entweder einen Wechsel der Ansicht oder bieten eine Funktion, wie zum Beispiel das Hinzufügen eines neuen Eintrages oder das Hochladen einer Datei an.

Um das Bearbeiten der einzelnen Tage im Berichtsheft einfacher zu gestalten, werden dem Benutzer in der Navigation Bar der Detailansicht Pfeile zum Blättern zwischen den Tagen angeboten. Dadurch muss der Benutzer nicht erst zurück zur Übersicht navigieren, um einen anderen Tag zu bearbeiten, sondern kann einfach und schnell zwischen den Tagen wechseln. Durch das Verwenden der Standardelemente von iOS wird dem Benutzer nicht nur eine vertraute Oberfläche geliefert, sondern auch die einfache und schnelle Erfüllung seiner Arbeitsaufgaben ermöglicht. Die Anwendung wirkt durch die Einteilung in die verschiedenen Ansichten und somit Inhaltsbereiche sehr aufgeräumt und übersichtlich.

5.3 Das Bedienkonzept für BLok

Das Bedienkonzept für die mobile Webanwendung und die native App soll die Steuerung mit Gesten verdeutlichen. Es wird darauf geachtet, nur einfache, den Benutzern bekannte Gesten zu verwenden. Dadurch wird sichergestellt, dass die Nutzer die Bedienung nicht erst erlernen müssen, sondern die Anwendung intuitiv steuern können. Weiterhin ist durch die Verwendung einfacher Gesten die Bedienung auch für Menschen mit Einschränkungen

der kognitiven Fähigkeiten leichter. In Abbildung 76 sind die für BLok vorgesehenen Gesten dargestellt.

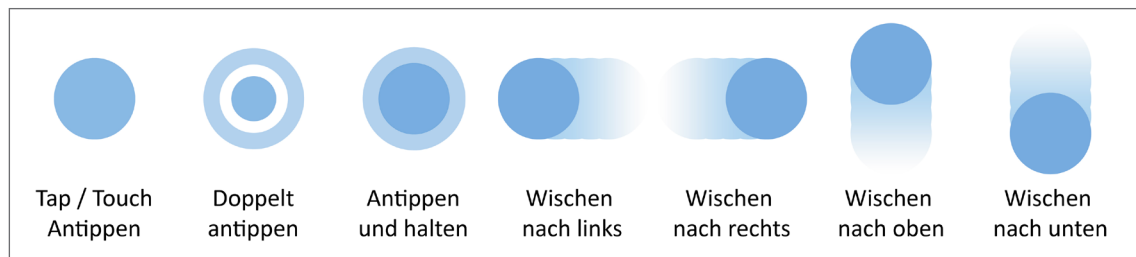


Abbildung 76 BLok: Verwendete Gesten

Neben dem einfachen Touch und dem Wischen in verschiedene Richtungen zum Scrollen oder Blättern ist im Rahmen der mobilen Webanwendung noch das Zoomen möglich. Für das Vergrößern des Inhaltes werden zwei Finger voneinander weg bewegt, für das Verkleinern entsprechend aufeinander zu. Alternativ kann diese Funktion auch durch ein schnell hintereinander ausgeführtes doppeltes Antippen ausgelöst werden.

Für die iOS App stehen die Gesten für das Zoomen nicht zur Verfügung, da der Inhalt genau an die Größe des Bildschirmes angepasst und somit das Vergrößern oder Verkleinern nicht notwendig ist. Dem Benutzer ist es aber trotzdem möglich, den Schriftgrad für eine bessere Lesbarkeit anzupassen. Der Schriftgrad kann über das Menü Einstellungen unter iOS verändert werden.

Für das Markieren und Korrigieren von Texten ist ein Antippen und halten notwendig. Dem Benutzer ist es somit jederzeit möglich, seine verfassten Texte auf den Buchstaben genau zu korrigieren.

5.3.1 Bedienung der mobilen Webanwendung

Die Bedienung der mobilen Webanwendung ist einfach gehalten und orientiert sich an den Konventionen des Internets. Der Benutzer muss die Gesten zur Bedienung daher nicht neu erlernen sondern kann seine Erfahrungen und Kenntnisse anwenden.

Ein einfaches Antippen öffnet neue Seiten, Links oder Elemente und funktioniert analog zum Klick mit einer Maus. Zu beachten ist allerdings, dass es bei der Bedienung mit den Fingern kein Mouseover bzw. Hover gibt. Inhalte sollten daher nie ausschließlich über Mouseover zugänglich sein. Für Benutzer einer Maus kann die Funktion erhalten bleiben, für Nutzer von mobilen Geräten sollten die Inhalte zum Beispiel durch einfaches Antippen sichtbar werden.

In Abbildung 77 ist die Bedienung der mobilen Anwendung verdeutlicht. Bei einem Tap auf den Button Menü wird zum Inhalt im unteren Bereich der Seite gescrollt. Der Benutzer kommt auf diesem Wege direkt zur Navigation. Die aktuelle Position, im Beispiel „Übersicht“, ist zur Orientierung hervorgehoben. Der Benutzer kann nun durch Antippen weitere Bereiche der Seite öffnen. Der ausgewählte Menüpunkt wird ebenfalls hervorgehoben, solange die neue Seite noch nicht geöffnet wird. Dies signalisiert dem Benutzer, dass die Anwendung seine Aktion verstanden hat und diese ausführt. Dauert diese Aktion unerwartet lange, sollte dem Benutzer ein Fortschrittsbalken angezeigt werden, um ihn darüber zu informieren.

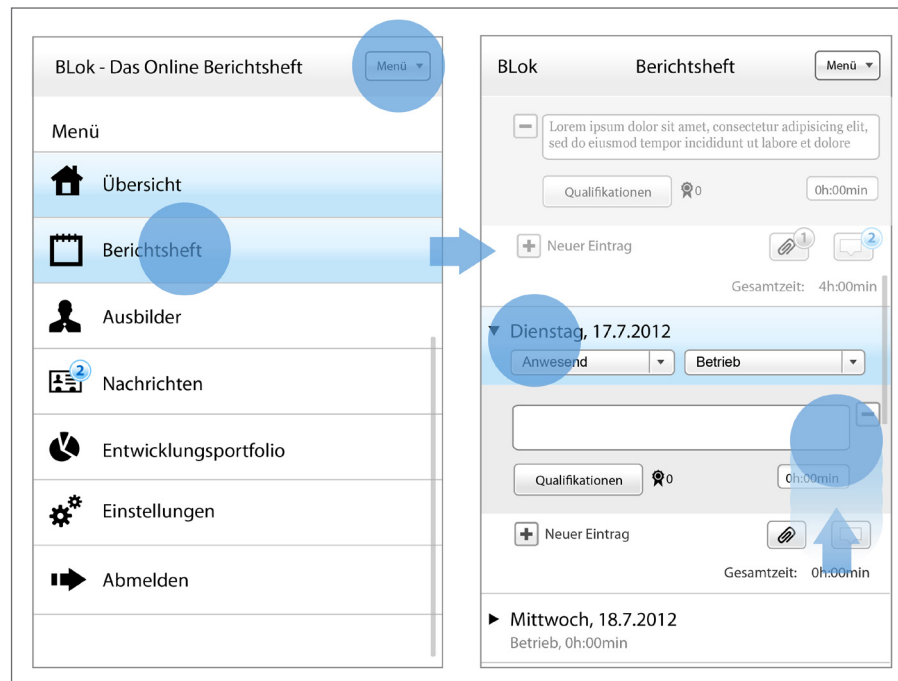


Abbildung 77 BLok: Bedienung der Webanwendung

Das Berichtsheft besteht aus einer Liste der Tage Montag bis Sonntag. Öffnet der Benutzer das Berichtsheft, ist der aktuelle Tag bereits aufgeklappt. Die anderen Tage bleiben aufgrund der Übersichtlichkeit geschlossen und zeigen nur die wichtigsten Informationen, wie die Anwesenheit oder neuen Kommentare an. Tippt der Benutzer einen geschlossenen Tag an, öffnet sich dieser. Gleichzeitig schließt sich der vorher geöffnete Tag. Der ausgewählte Tag wird wieder hervorgehoben, damit der Benutzer ein Feedback für seine Aktion bekommt.

Eine weitere Geste ist das Scrollen nach oben oder unten, um weitere Inhalte der Seite anzeigen zu lassen. Durch einen Scrollbalken wird die aktuelle Position des Benutzers auf der Seite angezeigt.

Für das Korrigieren des Textes stehen die Gesten „Doppelt Antippen“ sowie „Antippen und Halten“ zur Verfügung. Die Funktion sowie die Reaktion auf diese Gesten sind allerdings innerhalb der Systeme und der Browser unterschiedlich. Da das Korrigieren von Texten für BLok sehr wichtig ist, sollten die Gesten auf mehreren Endgeräten mit verschiedenen Browsern getestet werden um die Funktionsfähigkeit sicherzustellen.

In Safari unter iOS funktioniert das Korrigieren von Texten mit diesen Gesten sehr gut. Durch doppeltes Antippen wird das aktuelle Wort bzw. der gesamte Inhalt eines Textfeldes markiert. Die Funktionen Ausschneiden und Kopieren stehen dem Benutzer zur Verfügung. Außerdem kann er, falls die Rechtschreibprüfung aktiviert ist, Vorschläge für die korrekte Schreibweise abrufen. Durch die Geste „Antippen und Halten“ wird eine Lupe aktiviert, die das genaue Positionieren des Cursors ermöglicht. Der Benutzer kann dadurch einzelne Buchstaben korrigieren.

5.3.2 Bedienung der iOS App

Zwischen der Bedienung der Webanwendung und der nativen App für iOS gibt es zahlreiche Gemeinsamkeiten. Größtenteils arbeitet der Benutzer mit einfachem Antippen und Wischen um die App zu steuern und durch diese zu navigieren. Abbildung 78 zeigt einige Gesten und deren Resultat.

Unter iOS dient die Listenansicht unter anderem der Navigation. Enthält ein Eintrag einen Pfeil auf der rechten Seite, steht eine weitere Ansicht im Zusammenhang mit dem Listenelement zur Verfügung. Durch ein einfaches Antippen kann ein Element markiert werden. Als Feedback wird die Auswahl hervorgehoben und die nächste Ansicht von rechts nach links ins Bild geschoben. Über den Zurück-Button im Kopfbereich kann der Nutzer jederzeit zur letzten Ansicht zurückkehren.

Auch innerhalb der iOS App wird durch Wischen nach oben oder unten gescrollt um weiteren Inhalt anzuzeigen. Unter iOS erscheint der Scrollbalken nur, wenn der Benutzer die Funktion tatsächlich nutzt. Ansonsten wird dieser aufgrund der Übersichtlichkeit ausgeblendet.



Abbildung 78 BLok: Bedienung der iOS App - Ansicht 1

Für das Bearbeiten und Korrigieren von Texten stehen unter iOS zahlreiche Funktionen zur Verfügung. Mit der Geste „Antippen und Halten“ kann der Cursor direkt im Text platziert werden. Dem Benutzer wird eine Lupe angeboten, die das exakte Positionieren ermöglicht. Der Benutzer kann einzelne Wörter oder den gesamten Text markieren, kopieren und ausschneiden sowie einfügen. Durch doppeltes Antippen wird direkt das ausgewählte Wort markiert. Weiterhin werden Vorschläge für Wörter bei einem Rechtschreibfehler angeboten. Mit einem Tap kann der Benutzer das Wort dann austauschen bzw. korrigieren.

In Abbildung 79 ist beispielhaft eine Anwendungsmöglichkeit der Geste „Wischen nach rechts bzw. links“ zu sehen. Auf diesem Weg kann zum Beispiel eine Funktion aktiviert oder deaktiviert werden, indem der Schieberegler bearbeitet wird. Das Wischen nach rechts oder links kann außerdem zum horizontalen Scrollen bzw. Blättern verwendet werden.

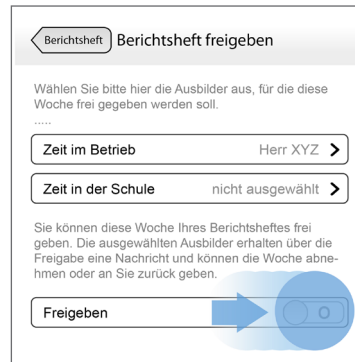


Abbildung 79 BLok: Bedienung der iOS App - Ansicht 2

Grundsätzlich navigiert der Benutzer innerhalb der iOS App von Ansicht zu Ansicht. Wird zum Beispiel eine Liste als Menü genutzt, schieben sich die Ansichten jeweils von links oder rechts ins Bild. Die Navigation ist demnach eher horizontal angelegt. Für Aktionen, die nur eine Ansicht betreffen, wird die dazugehörige sogenannte „Modal View“ (Modal-Dialog) von unten nach oben eingeblendet und entsprechend in die Gegenrichtung wieder ausgeblendet. Im Rahmen von BLok wird diese Funktion unter anderem für das Hochladen von Dateien, das Verfassen von Kommentaren oder das Anlegen eines neuen Beitrages innerhalb eines Tages verwendet.

Eine Modal View kann sich außerdem auf der Rückseite der aktuellen Ansicht befinden. Dies ist zum Beispiel bei den Einstellungen die gesamte Anwendung betreffend der Fall. Wird der Button aktiviert, dreht sich die aktuelle Ansicht auf die Rückseite und zeigt die Optionen an. Durch das Bestätigen oder Abbrechen wird die Ansicht wieder zurück auf die Vorderseite gedreht.

6 Gestalterische und technische Umsetzung der Anwendung BLok

Das Konzept, welches in Kapitel 5 vorgestellt wurde, dient als Grundlage für die gestalterische und technische Umsetzung der Anwendung BLok. Als Richtlinie für die Gestaltung der Benutzeroberfläche wurde zunächst ein Styleguide entwickelt. Entsprechend dieser Richtlinien und der Entwürfe für die Webanwendung sowie die mobile Anwendung erfolgt die gestalterische und technische Umsetzung.

6.1 Gestalterische Umsetzung

6.1.1 Entwicklung eines Styleguides für BLok

Ein Styleguide für ein Produkt oder eine Anwendung fasst Gestaltungsrichtlinien und verbindliche Vorgaben zusammen, die dem Designer bei der Umsetzung helfen. Die Richtlinien sollten unter anderem Farbdefinitionen, das Logo und dessen Einsatz sowie typografische Vorgaben enthalten. Der Styleguide für die Anwendung BLok ist in Anlage C zu finden.

Für BLok existieren bereits einige Vorgaben wie das Logo, welches in Abbildung 80 zu sehen ist, sowie das Farbschema. Da BLok bereits seit einiger Zeit im Raum Deutschland verbreitet ist, werden sowohl das Logo als auch das charakteristische Grün beibehalten. Die Benutzer und Kunden sind mit diesen Komponenten bereits vertraut und assoziieren die Anwendung BLok mit diesen. Eine komplette Neugestaltung der Außendarstellung ist somit nicht vorteilhaft.

Zusätzlich zum bereits bestehenden Logo wurde eine reduzierte Variante erstellt. Diese kommt ohne die unterschiedliche Färbung der Buchstaben, deren Positionierung sowie der Spiegelung aus. Der Schriftzug „BLok“ ist im charakteristischen dunklen Grün gehalten und der Zusatz „Das Online-Berichtsheft“ befindet sich darunter. Zusätzlich zu dieser Variante wurde ein Entwurf des Logos mit weißer Schrift auf grünem Hintergrund hinzugefügt. Beide Varianten sind als Ergänzung im Styleguide in Anlage C zu finden.



Abbildung 80 Logo BLok - Das Online-Berichtsheft

In Abbildung 81 ist das neue Farbschema der Anwendung BLok dargestellt. Neben den bereits verwendeten Grüntönen werden für die primäre Gestaltung warme Brauntöne und Grau verwendet. Das eher dezente Braun dient als Kontrast für das vordergründige Grün und soll dieses an einigen Stellen innerhalb der Anwendung ablösen. Die Grautöne werden weiterhin für die typografische Gestaltung verwendet.

Die sekundären Farbtöne Orange, Rot und Blau werden zur Auflockerung, für die Gestaltung von Hinweisen und zur Kennzeichnung der Benutzerrollen eingesetzt.

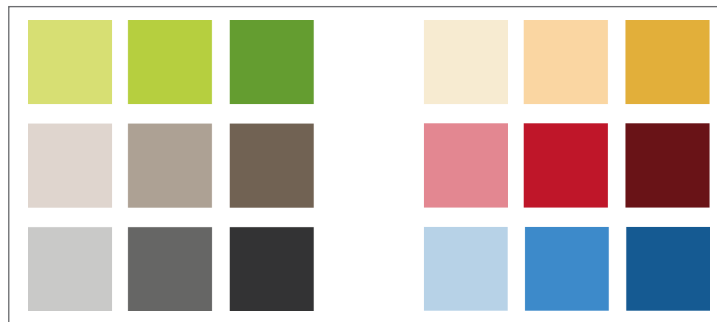


Abbildung 81 BLok: Primäre und sekundäre Farben

Im Rahmen der typografischen Gestaltung wurde die Schriftart Droid Sans ausgewählt. Bei dieser handelt es sich um eine serifenlose Schrift, die sowohl bei kleinem als auch bei großem Schriftgrad gut am Bildschirm lesbar ist. Weiterhin werden drei Ebenen für Überschriften definiert. Die Überschrift erster Ebene wird im charakteristischen Grün von BLok gehalten und betitelt jeweils die aktuelle Ansicht der Anwendung. Die weiteren Ebenen sind für die detaillierte Strukturierung des Inhaltes notwendig und werden in Grau gehalten. Mehr als drei Ebenen werden aufgrund der Übersichtlichkeit und Nachvollziehbarkeit für den Benutzer nicht verwendet.

Für die Gestaltung von Links wird ebenfalls der dunkle Grünton genutzt. Beim Darüberfahren mit der Maus oder Fokussieren des jeweiligen Elementes erscheint zusätzlich eine Unterstreichung um die Interaktionsmöglichkeit zu verdeutlichen. Links sollten für Aufrufe von internen oder externen Seiten verwendet werden.

Buttons für Funktionen wie zum Beispiel Speichern oder Abbrechen werden in einem dezenten Grauton gestaltet, der bei Erhalt des Fokus dunkler wird. Besonders wichtige Buttons können außerdem mit Grün hervorgehoben werden. Die auf diese Weise ausgezeichneten Schaltflächen können den Benutzer weiterhin auf noch nicht gespeicherte Änderungen aufmerksam machen.

Bei einigen Interaktionselementen ist ein zusätzlicher Hinweis auf die Funktion für das Verständnis wichtig. Das betrifft zum Beispiel das Anhängen von Dateien, die Möglichkeit, Kommentare zu verfassen oder auch das Registrieren. Aufgrund dessen bekommen diese Funktionen jeweils ein Icon. Neben dem schnelleren Verständnis kann mit den Icons die Aufmerksamkeit des Benutzers bewusst angezogen werden. Weiterhin helfen diese Bilder Benutzern, die Schwierigkeiten mit dem Verstehen von Texten haben. Außerdem könnte der Text, der die Funktion beschreibt, auch entfallen, wodurch Platz eingespart wird. Dies sollte aber nur bei eindeutigen und aussagekräftigen Symbolen wie einem Plus oder einem Fragezeichen für die Hilfe durchgeführt werden.

In Abbildung 82 sind sowohl die Icons für die Funktionen, als auch die Symbole für die Hauptnavigation zu sehen. Für die Navigation werden vier unterschiedliche Status verwendet. Neben der normalen Anzeige werden die Symbole bei Erhalt des Fokus heller eingefärbt sowie die nicht verfügbaren Funktionen grau dargestellt. Aktive Menüpunkte werden mit Hilfe der Farbe Grün hervorgehoben.



Abbildung 82 BLok: Icons für die Navigation und für ausgewählte Funktionen

6.1.2 Gestaltung der Webanwendung

Entsprechend der Vorgaben und Richtlinien, die im Konzept aus Kapitel 5 und im Styleguide für BLok zu finden sind, werden Layout-Entwürfe für die Webanwendung und für die mobile Version erstellt.

In Abbildung 83 ist die Gestaltung des Kopfbereiches und der Navigation zu sehen. Hauptsächlich wird das für BLok charakteristische dunkle Grün verwendet, da diese Farbe bereits durch das bestehende Logo vorgegeben ist. Zusätzlich wird ein warmes Braun zur Auflockerung eingesetzt, welches zum Beispiel bei den Icons der Hauptnavigation zu sehen ist.

Im Header ist außerdem das Profilbild des Benutzers platziert. Die Farbe der Umrahmung entspricht dabei der Rolle des Benutzers. An Stelle des Platzhalters kann der Benutzer ein individuelles Profilbild hinterlegen.

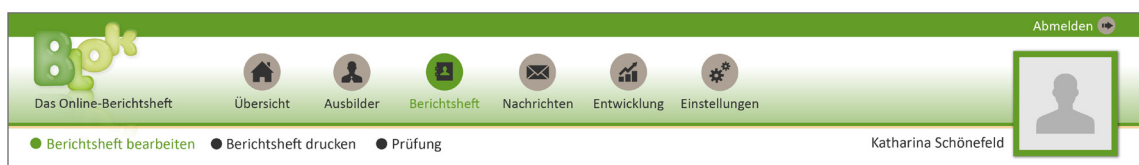


Abbildung 83 BLok: Layout Kopfbereich und Navigation

Der Kopfbereich und die Navigation werden für die mobile Version der Seite entsprechend reduziert. Trotzdem muss der Benutzer eine Verbindung zwischen den beiden Versionen herstellen können, sodass die Gestaltungsmerkmale größtenteils erhalten bleiben sollten.

Der Entwurf des Headers ist in Abbildung 84 dargestellt. Das Logo wurde an die verringerte Höhe angepasst und kleiner dargestellt. Des weiteren wird der Schriftzug „Das Online-Berichtsheft“ neben das Logo gesetzt. Außerdem muss im Header der Button zum Aufruf des Menüs platziert werden. Dieser wird entsprechend des Styleguides als Icon mit Text dargestellt.

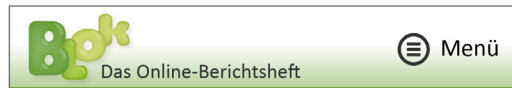


Abbildung 84 BLok: Entwurf Kopfbereich der mobilen Version

Die Hauptnavigation, die in Abbildung 85 zu sehen ist, wird zunächst ausgeblendet und kann über den Button Menü im Header aufgerufen werden. Als klare Abgrenzung zum restlichen Inhalt und als Blickfang wird die Überschrift grün hinterlegt. Die Elemente sind in einer Liste angeordnet und mit den Symbolen versehen. Die aktuelle Position des Benutzer ist durch die grüne Farbgebung gekennzeichnet.

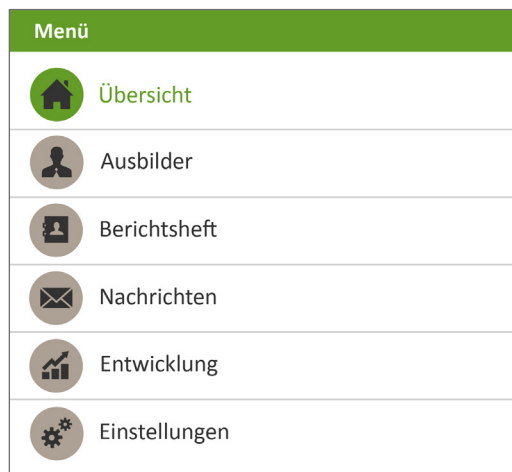


Abbildung 85 BLok: Entwurf Navigation der mobilen Version

Für die Darstellung von Informationen werden dezente und einfache Elemente eingesetzt. Dies wird zum Beispiel am Entwurf der Startseite deutlich, die in Abbildung 86 zu sehen ist. Dem Benutzer werden allgemeine Informationen sowie Neuigkeiten zum Thema BLok angeboten. Weiterhin erhält er Auskunft über die Vollständigkeit seines Berichtsheftes und über neue Benachrichtigungen. Die einzelnen Bereiche werden mit Hilfe von Linien, Abständen und Überschriften voneinander abgegrenzt. Durch die Anordnung von drei nebeneinander liegenden Bereichen wird der Platz optimal ausgenutzt, sodass der Benutzer ohne Scrollen bereits die wichtigen Inhalte erfassen kann.

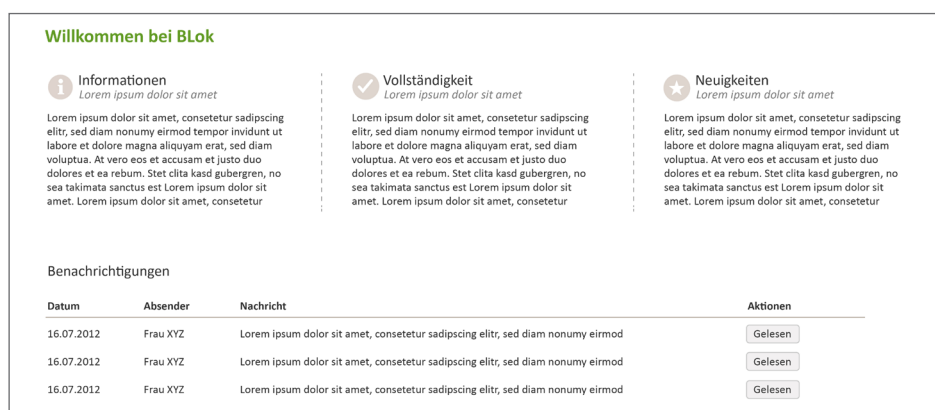


Abbildung 86 BLok: Gestaltung der Startseite

Bei der mobilen Version der Startseite werden die Inhalte aufgrund des beschränkten Platzes untereinander angeordnet. Das Wichtigste sollte so weit wie möglich oben platziert werden, damit der Benutzer auch im mobilen Kontext ohne Scrollen diese Inhalte sehen kann.

Auch das Berichtsheft wird übersichtlich und mit einfachen Mitteln gestaltet. Linien, Rahmen, farbige Flächen und Abstände sowie Überschriften dienen der Abgrenzung von verschiedenen Bereichen. Die einzelnen Tage werden zum Beispiel umrahmt und mit ausreichend Abstand versehen, um die Bereiche eindeutig voneinander zu trennen. Die Überschrift der Tage, also der entsprechende Wochentag und das Datum, werden farblich hervorgehoben und somit dominanter präsentiert, sodass der Benutzer schnell den richtigen Bereich finden kann. Zusätzlich zur braunen Hintergrundfarbe wird unterhalb eine Linie hinzugefügt, die dem aktuellen Aufenthaltsort des Auszubildenden entspricht. Wählt der Benutzer in der Auswahlliste Betrieb, so ist die Umrahmung zum Beispiel in Blau gehalten, wie auch in Abbildung 87 zu sehen. Die ausgewählten Farben entsprechen dabei den Benutzerrollen betrieblicher Ausbilder, schulischer Ausbilder und externer Ausbilder.

Abbildung 87 BLok: Gestaltung des Berichtshefts

In der mobilen Version des Berichtshefts muss die Gestaltung entsprechend an den beschränkten Platz angepasst werden. Der Entwurf für kleine Bildschirme ist in Abbildung 88 dargestellt. Gestaltungselemente wie Icons und Buttons werden wieder verwendet, um die Konsistenz zur Webanwendung zu wahren.

Der Kopfbereich mit dem Wochentag und dem Datum des ausgewählten Tages wird in der mobilen Version oberhalb angeordnet. Die Linie darunter kennzeichnet den jeweiligen Aufenthaltsort des Auszubildenden. Der Kopfbereich, der Inhaltsbereich und der Bereich für zusätzliche Optionen, wie zum Beispiel das Hinzufügen eines Kommentars, werden mit einer dezenten Hintergrundfarbe voneinander abgegrenzt.

Grundsätzlich entspricht die Reihenfolge und Anordnung der Elemente dem Entwurf der Webanwendung. Die Elemente sind nur nicht mehr nebeneinander angeordnet, sondern aufgrund des Platzes untereinander. Einige Funktionen, wie das Hinzufügen eines Kommentars, werden in der mobilen Version nur noch durch Icons dargestellt, um weiteren Platz zu sparen.

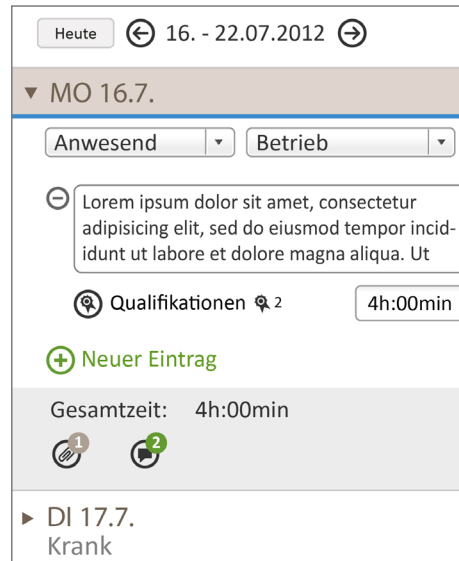


Abbildung 88 BLok: Berichtsheft in der mobilen Version

6.1.3 Gestaltung der nativen App

Bei der Gestaltung der nativen App für iOS ist die Konsistenz zu diesem System besonders wichtig. Weiterhin muss der Benutzer die App mit der Webanwendung BLok in Verbindung bringen können. Daher sind Logo, Farben und sonstige Gestaltungselemente in beiden Anwendungen konsistent zu nutzen. In Abbildung 89 ist der Entwurf für die Startseite und das Berichtsheft zu sehen.

Das dunkle Grün wird in der Navigation Bar, bei den Buttons und auch bei der Markierung von ausgewählten Einträgen verwendet. Für den Hintergrund und die Überschriften wird zur Auflockerung ein dezenter Braun-Ton eingesetzt.

Wie auch bereits im Konzept vorgesehen erfolgt die Navigation innerhalb der Anwendung über eine Table View. Diese Listenansicht ermöglicht neben der Navigation eine Übersicht über die verfügbaren Elemente, zum Beispiel die Tage im Berichtsheft. Zusätzlich können zu jedem Eintrag weitere Informationen wie die Anwesenheit, der Aufenthaltsort und die gebuchte Zeit angezeigt werden. Der aktuelle Tag wird außerdem mittels grüner Schriftfarbe hervorgehoben.

Für die Hauptnavigation auf der Startseite werden die bereits aus der Webanwendung bekannten Icons genutzt. Diese sind dem Benutzer vertraut und er erkennt somit die Ziele der einzelnen Navigationselemente wieder. Neue Nachrichten können weiterhin durch ein Icon gekennzeichnet werden.

Für Informationen und Neuigkeiten ist auf der Startseite grundsätzlich wenig Platz, da die Navigation für den Benutzer bereits ohne Scrollen zugänglich sein sollte. Daher werden über der Navigation nur die wichtigsten Informationen angezeigt, weitere Inhalte entsprechend darunter.

Auch das Profilbild des Benutzers wird im oberen Bereich der Startseite eingebunden. Dieses Element ist nicht im Header platziert, da nur sehr wenig Raum in diesem Bereich zur Verfügung steht. Neben dem Zurück-Button und der Überschrift der aktuellen Ansicht werden im

Header noch weitere Funktionen platziert. Dies ist zum Beispiel bei der Detailansicht des Tages der Fall. An dieser Stelle werden dem Benutzer Buttons angeboten, mit denen er direkt in der Ansicht zwischen den einzelnen Tagen blättern kann.

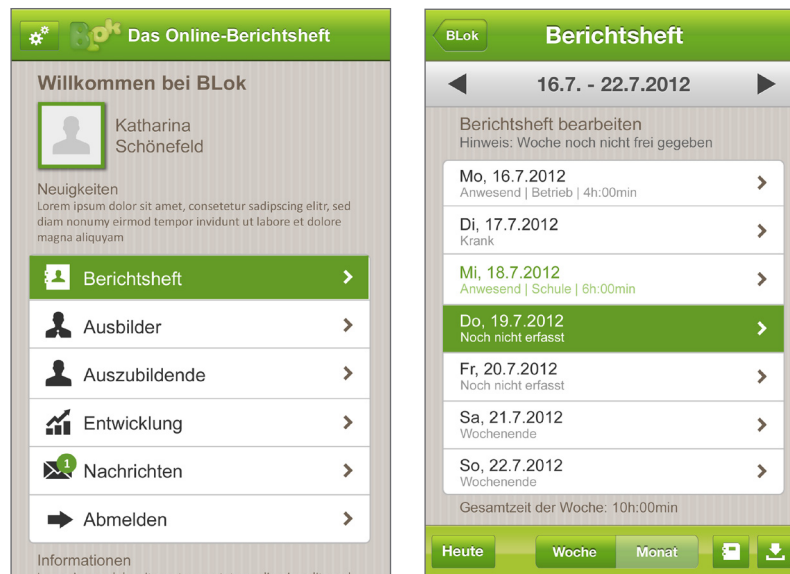


Abbildung 89 BLok: Gestaltung der App für iOS

6.2 Technische Umsetzung

6.2.1 Webanwendung BLok

Für BLok wird eine Webanwendung erstellt, die mit allen Geräten genutzt werden kann. Es wird demnach keine eigenständige, von der Webanwendung getrennte, mobile Variante geben. Grundsätzlich soll die Anwendung BLok für alle Benutzer den gleichen Informations- und Funktionsumfang bieten. Es soll keine Einschränkungen aufgrund der Fähigkeiten des Benutzers oder des Gerätes, welches der Benutzer verwendet, geben. Daher wurde die Erstellung einer separaten mobilen Anwendung zunächst ausgeschlossen.

Weiterhin ist ein wichtiger Grund der minimierte Pflegeaufwand, da nur für eine Anwendung neue Funktionen oder Informationen hinzugefügt werden müssen. Das Layout von BLok muss weiterhin auch für verschiedene Kunden anpassbar sein. Mit nur einer Anwendung kann eine neue Gestaltung sehr viel schneller umgesetzt werden.

Zunächst sollte weiterhin die Resonanz der Benutzer auf die mobile Version abgewartet werden, bevor zusätzlicher Aufwand in eine separate Anwendung oder sogar in verschiedene native Apps fließt. Nach der Fertigstellung von BLok sollten Benutzerbefragungen stattfinden und Statistiken über die Nutzung ausgewertet werden. Wenn sich zeigt, dass die Nutzung von BLok mit mobilen Geräten entsprechend hoch ist, kann erneut über die Erstellung einer separaten mobilen Anwendung oder sogar einer nativen App nachgedacht werden.

Die bisherige Anwendung BLok wurde nicht übernommen, sondern es wurde mit der Erstellung einer neuen Version begonnen. Dieses Vorgehen ermöglicht das Implementieren einer sauberen und aufgeräumten HTML-Struktur sowie der Aktualisierung der Technologie im Hintergrund.

6.2.1.1 HTML5

Die Anwendung BLok basiert auf der Technologie HTML5. Besonders die Sectioning Elemente wie `<header>` und `<nav>` wurden im Rahmen der technischen Umsetzung verwendet. An vielen Stellen konnte weiterhin die Syntax vereinfacht werden, wie zum Beispiel der Doctype und die Angabe der Metadaten. Da HTML5 in allen modernen Browsern verfügbar ist, waren nur kleine Anpassungen notwendig. Für den Internet Explorer ab Version 9 abwärts wurden daher die neuen Elemente über `document.createElement` bekannt gemacht und konnten so auch in älteren Versionen des Internet Explorers angesprochen werden.

Durch die Verwendung der HTML5 Sectioning Elemente erhält die Anwendung eine semantische Strukturierung. Auch Benutzer von assistiven Technologien profitieren von diesen Anpassungen, da Bereiche wie die Navigation oder der Footer als solche ausgezeichnet sind und entsprechend ausgegeben werden. In Abbildung 90 ist die grobe Struktur der Anwendung BLok mit den HTML5 Sectioning Elementen und der Javascript-Anweisung für die älteren Versionen des Internet Explorers dargestellt.

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="UTF-8" />
    <!--[if lt IE 9]>
      <script>
        document.createElement('section');
        document.createElement('nav');
        document.createElement('header');
        ...
      </script>
    <![endif]-->
  </head>
  <body>
    <header class="pageheader" role="banner">
      <h1>BLok - Das Online-Berichtsheft</h1>
      <nav class="mainnav" role="navigation">
        <ul>
          <li>Menü-Element</li>
        </ul>
      </nav>
    </header>
    <section class="content" role="main"></section>
    <footer class="footer" role="contentinfo"></footer>
```

Abbildung 90 BLok: Ausschnitt der HTML-Struktur

Die Technologien Offline Web Applications und Geolocation wurden noch nicht zu BLok hinzugefügt. Die Nutzung der Technologien ist aber zu empfehlen, da sich für den Benutzer daraus zahlreiche Vorteile ergeben. Die Auszubildenden können ihr Berichtsheft durch die mobile Verfügbarkeit von BLok auch von unterwegs aus pflegen. Es kann aber nicht sichergestellt werden, dass zu jeder Zeit eine Internetverbindung besteht. Trotzdem soll der Nutzer weiterhin Einträge verfassen und Speichern können. Zur Realisierung dieses Anspruches kann die Technologie Offline Web Applications verwendet werden.

Weiterhin ist es sinnvoll, die Möglichkeit der Lokalisierung über GPS zu nutzen. Der Benutzer könnte bestimmte Orte als Schule, Betrieb oder externe Einrichtung markieren, sodass der richtige Ort bei der Erstellung eines Eintrages bereits ausgewählt ist.

Beide Technologien werden bereits von allen modernen Browsern, außer Opera Mini unterstützt, sodass ein Einsatz empfohlen werden kann. Tests mit unterschiedlichen Browsern auf verschiedenen Geräten sollten zur Sicherstellung der korrekten Funktionsfähigkeit dennoch durchgeführt werden.

6.2.1.2 CSS

Grundsätzlich muss das CSS, mit dem die Anwendung BLok gestaltet wird, leicht zu warten und übersichtlich aufgebaut sein, da es jederzeit an individuelle Wünsche der Kunden anpassbar sein sollte. Daher bietet sich eine klare Gliederung der CSS-Datei an.

Für BLok wurde die CSS-Datei in verschiedene Bereiche eingeteilt sowie am Beginn der Datei ein kurzes Inhaltsverzeichnis hinterlegt. Der Inhalt wird unter anderem in Header, Navigation, Footer und Content strukturiert. Ein schnelles Erkennen der richtigen Anweisungen und die Übersichtlichkeit des Dokuments sind somit gewährleistet, sodass sich auch andere Bearbeiter zurechtfinden können.

Außerdem wurden die verwendeten Hexadezimal-Codes der Farben am Anfang der CSS-Datei aufgelistet, sodass diese einfach kopiert und schnell in eine Anweisung eingefügt werden können.

Für die Umsetzung wird zunächst nur eine CSS-Datei verwendet. Später sollten mindestens die unterschiedlichen Media Queries ausgelagert werden, um das Dokument übersichtlich zu halten. Layouts für Kunden werden in einer separaten Datei abgelegt, die auf dem Basis-Layout aufbaut und einige Anweisungen überschreibt.

Um die schnelle Anpassbarkeit an die Wünsche der Kunden zu demonstrieren, wurde beispielhaft ein Layout für Thyssen Krupp umgesetzt. Die Strukturierung der Anwendung sowie die Anpassungen an unterschiedliche Bildschirmgrößen müssen dabei nicht noch einmal vorgenommen werden. Stattdessen werden nur ausgewählte Anweisungen zum Austausch von Logo, Farben oder Schrift gezielt überschrieben. In Abbildung 91 sind zwei Ansichten des alternativen Layouts dargestellt.

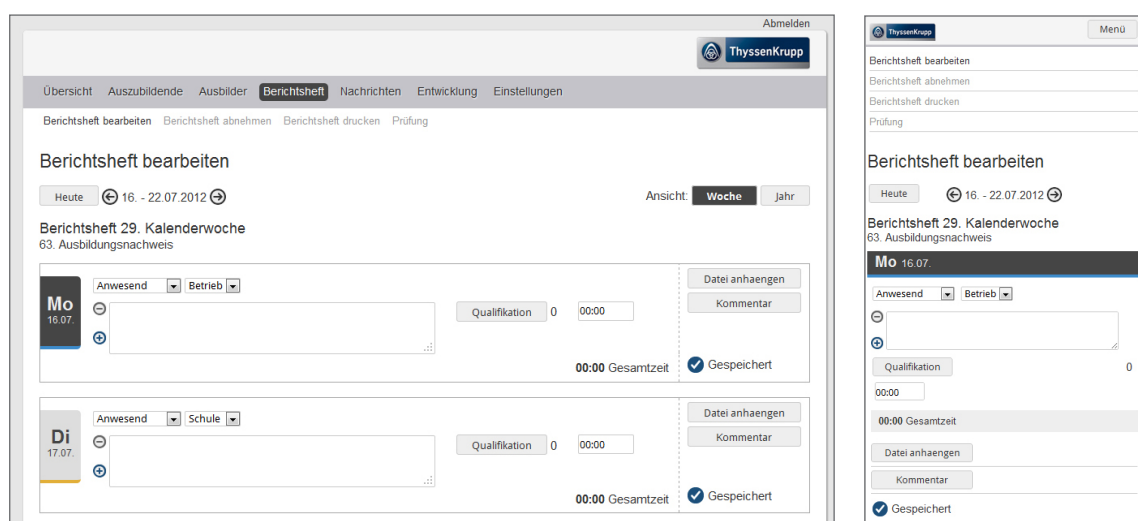


Abbildung 91 BLok: Angepasstes Layout für Thyssen Krupp

Alle notwendigen CSS-Dateien sollten abschließend im HTML über das `<link>` Element eingebunden werden. Auf diese Weise wird sichergestellt, dass nur die notwendigen Daten ausgeliefert werden. Außerdem ist die Verwendung von `<link>` zum Einbinden der Dateien performanter als das Verknüpfen mit der `@import` Anweisung in den CSS-Dateien.

6.2.1.3 WAI-ARIA

Grundsätzlich soll die Anwendung BLok für alle Menschen zugänglich sein, unabhängig von deren Fähigkeiten. Aus diesem Grund wird von Anfang an auf die Barrierefreiheit geachtet. Inhalte müssen für alle Nutzer zugänglich und verständlich sein. Daher sind für Medien alternative Inhalte zu hinterlegen sowie verständliche Texte zu formulieren. Neben den angepassten Inhalten ist auch die technische Umsetzung der Barrierefreiheit wichtig. Im Rahmen von BLok wird daher ARIA zur Auszeichnung der HTML-Elemente eingesetzt. Assistive Technologien können durch die ARIA-Attribute Bedeutungen und Zusammenhänge von Elementen an den Benutzer weitergeben.

Mit Hilfe der Rollen in ARIA bekommen Bereiche eine inhaltliche Bedeutung, die dem Benutzer ausgegeben wird. Auf diese Weise werden zum Beispiel die Navigation und der Inhaltsbereich definiert. Für den Benutzer sind die Inhalte dieser Bereiche besser verständlich und er hat weiterhin die Möglichkeit, schneller zu den einzelnen Bereichen zu navigieren. In Abbildung 90 ist der Einsatz der Rollen `banner`, `navigation`, `main` und `contentinfo` in der Anwendung BLok zu sehen.

ARIA wird außerdem verwendet, um Formulare besser zugänglich zu machen und dem Benutzer dynamische Änderungen des Inhaltes anzuzeigen. Dafür werden Zustände und Eigenschaften sowie die ARIA Live Regions verwendet.

Da ARIA keinen Einfluss auf das Layout oder die Funktionen einer Webanwendung hat, kann die Umsetzung dieser Technologie für BLok empfohlen werden. Somit wird eine gute Basis geschaffen, um erste Nutzertests durchzuführen. Sollten Benutzer mit physischen Einschränkungen BLok verstärkt nutzen, kann die Accessibility weiter ausgebaut werden.

6.2.2 Anpassungen für die mobile Webseite

Da für BLok eine Anwendung für alle Geräte erstellt wird, müssen für die mobile Variante einige Anpassungen vorgenommen werden. Diese beinhalten zum Beispiel die Anpassung des Layouts an die unterschiedlichen Bildschirmgrößen mit Hilfe der Technologien Responsive Webdesign und CSS3 Media Queries.

Weiterhin müssen einige Elemente für die mobile Version ausgetauscht oder neu positioniert werden. Dies soll mit serverseitiger Technologie umgesetzt werden.

6.2.2.1 Responsive Webdesign

Die Technologie Responsive Webdesign wird für die Anpassung des Layouts an verschiedene Bildschirmgrößen genutzt. Die Breakpoints orientieren sich aber nicht an vorgegebenen Maßen, wie zum Beispiel den Bildschirmgrößen des iPhones oder iPads, sondern werden abhängig vom Layout bestimmt.

Zu diesem Zweck wird zunächst die Desktop-Version der Anwendung zusammengestellt und mit Hilfe von CSS gestaltet. Anschließend wird das Browserfenster schrittweise verkleinert, um die Breakpoints zu finden. Es wird die Breite gesucht, ab der das Layout nicht mehr korrekt dargestellt wird. An dieser Stelle ist der erste Punkt für den Einsatz der Media Queries gefunden und das Layout kann korrigiert werden. Auf diesem Weg werden die jeweiligen Breakpoints gesucht, bis das Layout bei allen Bildschirmbreiten korrekt dargestellt wird. Für BLok wurden die in Abbildung 92 dargestellten Media Queries verwendet.

```
@media screen and (max-width: 1250px)
@media screen and (max-width: 999px)
@media screen and (max-width: 780px)
@media screen and (max-width: 655px)
@media screen and (max-width: 450px)
@media screen and (max-width: 290px)
```

Abbildung 92 BLok: Verwendete Media Queries

Bei einer Breite ab 1250 Pixel erfolgt die erste Korrektur des Layouts. Der gesamte Inhaltsbereich wurde auf 990 Pixel verkleinert, damit kein horizontaler Scrollbalken entsteht, sondern die gesamte Seite sichtbar bleibt. Außerdem wurden die drei nebeneinander liegenden Bereiche auf der Startseite schmaler gestaltet, um die Positionierung dieser beizubehalten.

Größere Anpassungen erfolgen ab einer Breite von 999 Pixeln. Damit Navigation und Profilbild weiterhin nebeneinander in den Header passen, wird das Logo verkleinert und über der Navigation platziert. Somit kann die Navigation bereits am linken Rand des Headers beginnen. Zum Vergleich ist in Abbildung 93 der Kopfbereich bei einer Breite von 1250 Pixeln und 999 Pixeln dargestellt.



Abbildung 93 BLok: Header bei 1250 Pixeln und 999 Pixeln Breite

Ab einer Breite von 655 Pixeln wird der Header stark reduziert und die Navigation in Listenform dargestellt. Zusätzlich erscheint der Menü-Button, über den die Navigation ein- und ausgeblendet werden kann. Bleibt die Navigation in dieser Position sichtbar, nimmt diese die gesamte Höhe des Smartphones ein, sodass der Inhalt für den Benutzer erst durch Scrollen erreichbar ist. Dies soll durch den Button vermieden werden, sodass der Inhalt bei Aufruf der Seite sofort zugänglich ist. Um für den Menü-Button Platz zu schaffen wurde weiterhin die Funktion Abmelden nach unten in die Navigationsliste verschoben.

Auch das Layout des Berichtsheftes passt sich an die verringerte Bildschirmbreite an. Der Kopfbereich wird zum Beispiel nicht mehr links neben dem Inhalt angezeigt, sondern befindet sich im oberen Bereich. Weiterhin werden der Button Qualifikation und das Eingabefeld für die Zeit unterhalb der Textbox platziert.

Die optionalen Funktionen, wie das Hochladen von Dateien, werden außerdem nicht mehr neben dem Hauptinhalt, sondern mit einer Linie getrennt darunter platziert. In Abbildung 94 sind Ausschnitte des Layouts bei einer Breite von 655 Pixeln und 450 Pixeln zu sehen.

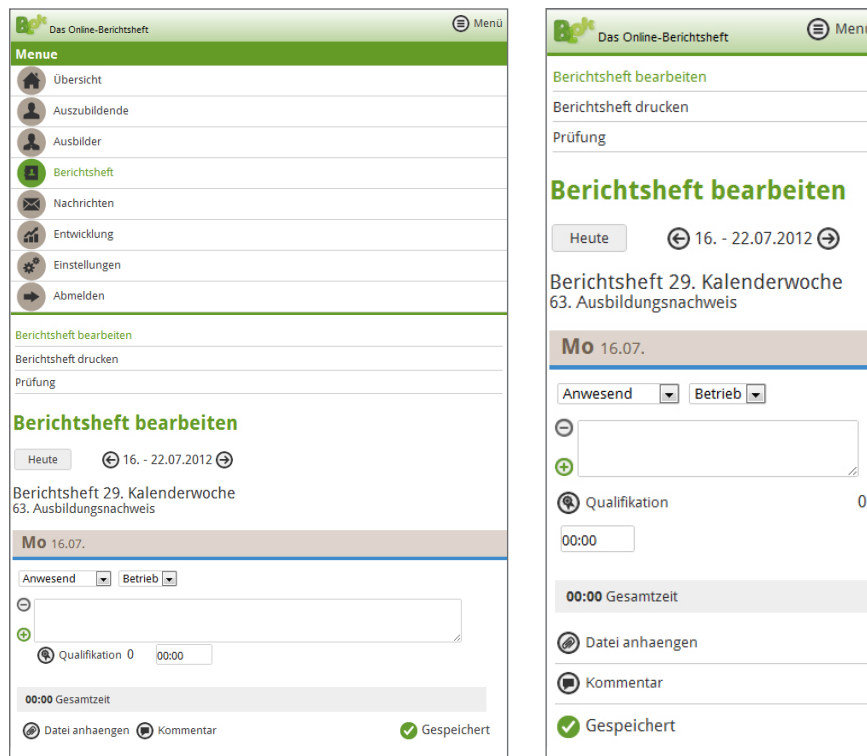


Abbildung 94 BLok: Layout bei 655 Pixeln und bei 450 Pixeln Breite

Da die CSS3 Media Queries nur in modernen Browsern funktionieren, wird auch nur in diesen das Layout an die Bildschirmgröße angepasst. In Internet Explorer Version 7 und Version 8 ist das Layout entsprechend nicht flexibel. An dieser Stelle muss entschieden werden, ob eine Anpassung auch an kleine Bildschirmgrößen bei diesen Versionen des Internet Explorers notwendig ist. Die mobilen Geräte nutzen zum Beispiel aktuelle Versionen von Safari, Chrome oder Opera und Windows Phone wird mit einer neuen Version des Internet Explorers ausgeliefert. Daher ist eine Anpassung prinzipiell nicht erforderlich.

Für eine Unterstützung der Media Queries in den älteren Versionen des Internet Explorers kann jederzeit Javascript genutzt werden. Respond.js und CSS3-MediaQueries.js bieten zum Beispiel in unterschiedlichem Funktionsumfang Lösungen an.²⁷⁵

Der Ansatz Mobile First, also das Beginnen mit der Umsetzung des mobilen Layouts, wurde für BLok nicht gewählt. Die Anwendung wurde für die Nutzung mit einem Desktop-PC optimiert und dann schrittweise an kleinere Bildschirmgrößen angepasst. BLok ist eine sehr umfangreiche Webanwendung mit vielseitigen Funktionen, die vordergründig für die Nutzung mit einem Desktop-PC gedacht ist. Die mobile Version der Anwendung ist zunächst eine Ergänzung der Nutzungsmöglichkeiten und ist im ersten Schritt für die Nutzergruppe Auszubildende gedacht, die auch unterwegs ihr Berichtsheft pflegen kann. Aus diesem Grund steht die mobile Version erst an zweiter Stelle, sodass der Ansatz Mobile First nicht geeignet ist.

275 Vgl. Nyman: Techniques For Gracefully Degrading Media Queries, 10.8.2011

6.2.2.2 *jQuery Mobile*

Das Javascript Framework jQuery Mobile wird im ersten Schritt der Umsetzung der Anwendung BLok nicht verwendet. Zum einen wurde für die Umsetzung von BLok die Technologie Responsive Webdesign gewählt, sodass eine Anwendung für alle Geräte entsteht. Um jQuery Mobile zu nutzen, wird aber größtenteils eine separate mobile Web App erstellt. Als Alternative müsste das mobile Layout auch für die Desktop-Webseite verwendet werden.

Zum anderen soll die Gestaltung für mobile Geräte nicht an eine native App erinnern. Der Benutzer arbeitet mit einer mobilen Webanwendung, die über den Browser aufgerufen wird und dies soll auch die Darstellung kenntlich machen. Durch das Framework jQuery Mobile entsteht allerdings eher ein natives Aussehen. Ein weiterer Nachteil eines unterschiedlichen Aussehens der mobilen Web App und der Webanwendung für Desktop-PCs ist die fehlende Konsistenz zwischen diesen. Dem Benutzer kann es dadurch unter Umständen schwer fallen, eine Verbindung zwischen den beiden Arten der Anwendung herzustellen.

Neben der Nutzung aufgrund gestalterischer Aspekte kann jQuery Mobile auch für Animationen sowie für spezielle Funktionen für Touchscreens verwendet werden. Zum Beispiel könnte das Menü nicht nur über einen Tap auf den Button im Kopfbereich aufgerufen werden, sondern auch durch eine Geste. Denkbar wäre das Einblenden durch ein Wischen von oben nach unten im oberen Bereich der Anwendung. Das Menü würde entsprechend der Geschwindigkeit und Position des Fingers bei der Geste immer weiter sichtbar werden.

Animationen sind weiterhin bei der Navigation insgesamt denkbar. Aktuell wird beim Aufruf eines Navigationspunktes die Seite neu geladen, sodass der Inhalt kurz verschwindet und anschließend neu aufgebaut wird. Sinnvoll kann an dieser Stelle das Hineinschieben der nächsten Seite von rechts nach links sein, so wie es auch bei nativen Apps verwendet wird. Solche Animationen sollten nicht nur aufgrund des gestalterischen Aspektes verwendet werden, sondern auch um dem Benutzer die Funktionen der Anwendung klarer zu machen bzw. um Feedback zu geben. Durch das Auswählen eines Navigationspunktes wird zu einer anderen Ansicht gewechselt. Dieser Wechsel wird durch das Schieben dieser Ansicht ins Bild deutlicher. Außerdem erhält der Benutzer sofort ein Feedback, dass die Aktion ausgeführt wird und nicht nur eine weiße Seite und das erneute Laden der Anwendung.

Der Einsatz von jQuery Mobile für Animationen oder die Unterstützung von Gesten kann demnach für die Anwendung BLok empfohlen werden. Sollte die Resonanz auf die mobile Version der Anwendung entsprechend groß sein, lohnen sich auch funktionale Anpassungen und das Unterstützen von Gesten durch jQuery Mobile.

6.2.2.3 *Serverseitige Anpassungen*

Die Verwendung von Responsive Webdesign mit serverseitiger Auslieferung bestimmter Teile des Quellcodes kann für die Anwendung BLok empfohlen werden. Zum Beispiel ist es sinnvoll, eine andere Navigation für den mobilen Kontext bereitzustellen. Auch das Hinzufügen oder Entfernen bestimmter Funktionen aufgrund des verwendeten Gerätes ist für BLok nützlich. Für die mobilen Geräte könnten auf diese Weise Funktionen wie die Lokalisierung des Standortes hinzugefügt werden. Der Code für diese Funktion wird in der Desktop-Variante nicht benötigt und demzufolge auch nicht an den Benutzer ausgeliefert.

Aktuell erfolgt die Anpassung an verschiedene Endgeräte ausschließlich über das Layout. Zunächst sollen unterschiedliche Bildschirmgrößen in der Gestaltung berücksichtigt und

Korrekturen entsprechend mit Hilfe von Responsive Design und Media Queries umgesetzt werden. Die funktionale Umsetzung sowie Ergänzungen des Funktionsumfanges für die mobile Version werden erst später vorgenommen.

6.3 Evaluation der neuen Anwendung BLok

Die Analyse der überarbeiteten Version von BLok kann nur unvollständig durchgeführt werden, da es sich bei der Anwendung aktuell nur um einen Prototyp handelt. Bisher wurde vordergründig das Layout umgesetzt, die Funktionalitäten fehlen noch zum großen Teil. Daher wird in diesem Kapitel nur die bisher umgesetzte Oberfläche bewertet. Eine Zusammenfassung der Analyse in tabellarischer Form ist in Anlage D zu finden.

6.3.1 Bewertung nach ISO 9241-110 Dialoggestaltung

6.3.1.1 Aufgabenangemessenheit

Bei der Überarbeitung der Benutzeroberfläche von BLok wurde besonders auf eine klare Struktur und einen einfachen sowie leicht verständlichen Aufbau geachtet. Die Reihenfolge der Elemente wurde besser an die Reihenfolge der Arbeitsschritte des Benutzers angepasst. Im Berichtsheft sind zum Beispiel die Pflichtangaben wie die Anwesenheit, die Qualifikationen und die Erfassung der Zeit besser von den optionalen Angaben getrennt. Diese optionalen Funktionen umfassen die Möglichkeit Dateien anzuhängen und Kommentare zu verfassen. Weiterhin bleibt Raum für Informationen, zum Beispiel ob der aktuelle Tag bereits gesichert wurde.

Durch das Ausblenden der Hauptnavigation bei kleinen Bildschirmen steht außerdem stets der Inhalt im Vordergrund. Der Benutzer kann bereits ohne Scrollen einen Großteil des Inhaltes sehen und direkt bearbeiten. Die Hauptnavigation kann über den Button Menü eingeblendet werden, sodass diese nur verfügbar ist, wenn der Benutzer sie benötigt.

Um den Benutzer bei dem Ausfüllen von Formularelementen zu unterstützen, sind in Auswahllisten bereits typische Inhalte vorgegeben. Anstatt in diesen zum Beispiel „Bitte auswählen“ zu hinterlegen werden Anwesenheit, Aufenthaltsort oder verschiedene Ausbilder bereits ausgewählt. Der Benutzer muss die Auswahl nur anpassen, wenn er die Vorgabe nicht verwenden will.

6.3.1.2 Selbstbeschreibungsfähigkeit

Der Benutzer kann sich innerhalb der Anwendung über das Menü und die Überschriften orientieren. In der Version für kleinere Bildschirme wird das Hauptmenü allerdings ausgeblendet und steht somit nicht zu jeder Zeit zur Verfügung. Der Benutzer muss die Navigation über den Button Menü aufrufen um sich auf diese Weise zu orientieren. Die Unternavigation sowie die Hauptüberschrift des Inhaltsbereiches können zwar auch zur Orientierung verwendet werden, besser wäre aber auch das Zeigen der aktuellen Position bezogen auf die Hauptnavigation. Dazu sollte im Kopfbereich statt dem Namen der Anwendung der Titel der aktuellen Ansicht angezeigt werden.

Um die Anwendung BLok selbsterklärender zu gestalten, wurden weiterhin zahlreiche Beschriftungen eingefügt. Somit sind nicht nur alle Formularfelder mit einer Bezeichnung versehen, sondern auch zahlreiche Icons. Das Verständnis dieser Funktionen wird dadurch sehr viel leichter, da bei schwierigen Funktionen ein Icon alleine nicht ausreicht. Dies ist zum Beispiel bei den Qualifikationen der Fall. Durch die Beschriftung zusätzlich zum Icon wird die Bedeutung unmissverständlich.

6.3.1.3 Erwartungskonformität

Die innere Konsistenz von BLok wird stets gewahrt, da Anpassungen an verschiedene Bildschirmgrößen nur über das Layout stattfinden und nicht über separate Anwendungen realisiert werden. Das Logo, die Farben, die Icons und die Schriftgestaltung bleiben für alle verschiedenen Ansichten der Anwendung gleich. Durch die Verwendung des bereits bekannten Logos und der von den Benutzern mit BLok verbundenen Farbe grün bleibt der Wiedererkennungseffekt erhalten.

Durch das Einhalten von Konventionen zum Beispiel bei der Gestaltung von Buttons, Links oder Überschriften wird auch die Anforderung nach äußerer Konsistenz eingehalten.

Rückmeldungen für den Benutzer bei Problemen, Fehlern oder nur als Information sind im Konzept von BLok fest eingeplant. Zum Beispiel wird stets auf den Status einer Woche im Berichtsheft hingewiesen. Ausbilder können somit sofort sehen, ob eine Woche noch in Bearbeitung oder bereits durch den Auszubildenden freigegeben ist. Der Auszubildende bekommt wiederum einen Hinweis, ob sein Berichtsheft akzeptiert oder abgelehnt wurde. Die Art der Meldung wird durch die Farbe und ein Symbol gekennzeichnet. Neutrale Hinweise sind demnach in Blau gehalten und mit einem Informationssymbol versehen. Fehler werden dagegen rot und auffälliger gestaltet.

6.3.1.4 Lernförderlichkeit

Eine kontextsensitive Hilfe ist bereits in der vorherigen Version von BLok ein fester Bestandteil der Anwendung. Direkt bei Funktionen, für die Erklärungen und Hinweise notwendig sind, wird die Hilfe über ein Symbol angeboten. Bewegt der Benutzer die Maus über dieses Symbol, werden ihm weitere Informationen angezeigt. Die kontextsensitive Hilfe ist auch für die überarbeitete Anwendung BLok im Konzept mit vorgesehen und kann größtenteils übernommen werden. Zu beachten ist allerdings, dass auf mobilen Geräten mit Touchscreen die Funktion Mouseover größtenteils nicht zur Verfügung steht. Als Alternative sollte die Information beim Tap auf das Symbol eingeblendet werden.

Durch das unaufdringliche Anbieten der Hilfe wird der Benutzer gut beim Erlernen der Anwendung unterstützt. Für Benutzer, die weitere Informationen benötigen, sind diese schnell zugänglich. Fortgeschrittene Benutzer werden durch das platzsparende Symbol nicht unnötig abgelenkt.

Weiterhin kann der Benutzer die Anwendung BLok frei ausprobieren. Eingaben und ausgewählte Optionen können jederzeit bearbeitet und korrigiert werden. Vor schwerwiegenden Funktionen wie zum Beispiel dem Löschen von Einträgen muss der Benutzer gewarnt werden und die Aktion bestätigen. Dieser Arbeitsschritt ist im Konzept mit vorgesehen, muss aber noch technisch umgesetzt werden.

6.3.1.5 Steuerbarkeit

Der Benutzer kann die Geschwindigkeit und die Reihenfolge der Arbeitsschritte stets selbst bestimmen. Daher ist eine flexible und an die Bedürfnisse des Benutzers angepasste Verwendung der Anwendung BLok möglich.

BLok kann sowohl mit der Maus, als auch ausschließlich mit der Tastatur bedient werden. Alle Elemente, mit denen eine Interaktion möglich ist, sind daher über die Tastatur ansteuerbar. Erhalten diese Elemente den Fokus, werden sie durch eine Umrahmung hervorgehoben, sodass der Benutzer seine aktuelle Position innerhalb der Seite besser erkennt. Nachdem alle Funktionalitäten umgesetzt sind, sollte ein erneuter Test der Steuerbarkeit durchgeführt werden. Auf diese Weise wird sichergestellt, dass auch die hinzugekommenen Elemente und Interaktionen mit der Tastatur bedient werden können.

Außerdem kann die Bedienung der Anwendung auch über den Touchscreen von mobilen Geräten erfolgen. Durch die Verwendung von Links, Buttons und Formularelementen kann der Benutzer wie mit der Maus mit dem Finger Elemente auswählen und somit die Interaktionen aktivieren. Auch an dieser Stelle muss nach der Umsetzung aller Funktionen die Steuerung erneut kontrolliert werden. Besonders auf Inhalte, die nur per Mouseover zugänglich sind, ist zu achten. Bei den meisten Touchscreens ist diese Funktion nicht verfügbar, sodass die Inhalte auch per Tap geöffnet werden müssen. Inhalte dieser Art sind zum Beispiel Tool-tips, die unter anderem bei der kontextsensitiven Hilfe eingesetzt werden.

Neben der Maus, der Tastatur und dem Touchscreen muss eine Anwendung auch mit Hilfe von assistiven Technologien steuerbar sein. Ein Test mit dem Screenreader VoiceOver unter iOS wird in Kapitel 6.3.3 durchgeführt.

6.3.1.6 Fehlertoleranz

Um Fehler grundsätzlich zu vermeiden werden Hinweise und Meldungen angeboten sowie Pflichtfelder in Formularen gekennzeichnet. Durch eindeutige Icons und aussagekräftige Beschriftungen sollen Fehler bei der Bedienung weiterhin vermieden werden.

Fehler sind allerdings nicht immer vermeidbar, sodass in BLok verständliche und konstruktive Fehlermeldungen verwendet werden sollten. Weiterhin muss der Benutzer beim Erkennen von Fehlern unterstützt werden. Fehlerhafte Eingaben sollten daher markiert werden und den Fokus erhalten, damit der Nutzer ohne Umwege mit der Korrektur beginnen kann.

Das Konzept Fehlertoleranz wurde in BLok mit aufgenommen und die notwendigen gestalterischen Anpassungen sind bereits vorbereitet. Nach der funktionalen Umsetzung sollte das Kriterium nochmals untersucht werden.

6.3.1.7 Individualisierbarkeit

Bereits in der aktuellen Version von BLok können die Benutzer individuelle Einstellungen wie das Hinterlegen eines Profilbildes vornehmen. Neben dem Profilbild sind weitere Optionen für das Berichtsheft und die Benachrichtigungsfunktion verfügbar. Der Benutzer kann unter anderem entscheiden, ob im Berichtsheft das Wochenende mit angezeigt werden soll oder die Anzahl der Wochenstunden festlegen. Diese persönlichen Einstellungen sollen auch in der überarbeiteten Anwendung verfügbar sein.

Neben diesen Möglichkeiten können außerdem an die Kunden angepasste Layouts zur Verfügung gestellt werden. Kammern oder Unternehmen können somit für ihre Ausbilder und Auszubildende ein alternatives Layout aktivieren, welches zum jeweiligen äußeren Erscheinungsbild passt.

6.3.2 Bewertung Usability für mobile Geräte

6.3.2.1 Konsistenz

Da BLok als Webanwendung umgesetzt wurde, kann nur die Konsistenz der mobilen Version und der Version für große Bildschirme bewertet werden. Durch die Verwendung der gleichen Farben, Symbole und Schriftgestaltung bleibt die Konsistenz grundsätzlich gewahrt. Sonstige Ausführungen wurden bereits in Kapitel 6.3.1.3 Erwartungskonformität besprochen.

6.3.2.2 Navigation

Die Anwendung BLok ist sehr umfangreich, sodass viele Einträge in der Hauptnavigation zu finden sind. Daher ist die empfohlene Beschränkung auf drei bis vier Links nur schwer umsetzbar. In der mobilen Version sollen keine Seiten ausgeblendet werden, sodass dem Benutzer der gesamte Funktionsumfang zur Verfügung steht. Damit die umfangreiche Navigation nicht den gesamten Platz im oberen Bereich einnimmt, wird diese erst beim Aktivieren des Buttons im Menü eingeblendet. In der Unternavigation gibt es grundsätzlich nur maximal vier Links, sodass an dieser Stelle die Anforderung eingehalten wird.

Eine am unteren Bildschirmrand fixierte Navigation wird momentan für die mobile Version von BLok nicht genutzt. Diese Art der Umsetzung wäre aber für die Unternavigation denkbar. Durch die Beschränkung auf maximal vier Menüpunkte ist der Platz für eine Anordnung nebeneinander ausreichend. Auf diese Weise würde im oberen Bereich weiterer Platz gespart werden und die Navigation ist für den Benutzer von jeder Position auf der Seite aus erreichbar.

6.3.2.3 Informationsdarstellung

Durch das Ausblenden der Hauptnavigation und die Reduktion des Kopfbereiches der Anwendung wird bereits ein Großteil des Inhaltes angezeigt, ohne dass der Benutzer scrollen muss. Weitere Optimierungen wie das Platzieren des Titels der aktuellen Seite im Header stehen noch aus. An dieser Stelle ist eine Anpassung der Struktur der Anwendung notwendig. Die Hauptüberschrift einer Seite muss über der Navigation im Header platziert werden, sodass diese auch beim Wechsel einer Seite zuerst von einem Screenreader ausgegeben werden kann.

Eine weitere Anpassung für kleine Bildschirme ist das Ausnutzen der gesamten Breite. Daher wurden die seitlichen Abstände der Elemente stark reduziert, sodass der gesamte zur Verfügung stehende Platz verwendet wird. Auch Links und Buttons nehmen jeweils eine eigene Zeile ein und der aktive Bereich wird auf die gesamte Breite ausgedehnt. Dadurch fällt es dem Benutzer sehr viel leichter, diese Elemente mit den Fingern auszuwählen.

6.3.2.4 Interaktionen

Bei der Bedienung eines Touchscreens steht besonders die Größe der Interaktionselemente im Vordergrund. Da an dieser Stelle nicht der Mauszeiger, sondern der Finger verwendet wird, müssen die Elemente sowie deren Abstand zueinander entsprechend angepasst werden. Die Größe der Elemente bzw. des aktiven Bereiches dieser sollte daher ca. einen Zentimeter betragen, um ein fehlerfreies Auswählen zu gewährleisten. Bei der Anwendung BLok wurde auf diese Vorgabe geachtet, sodass die Bedienung des Touchscreens für den Benutzer so einfach wie möglich ist. Einige Icons für das Auslösen von Funktionen sind aus Platzgründen allerdings kleiner gehalten. Durch einen ausreichenden Abstand zum nächsten Interaktionselement sollen Fehler bei der Bedienung trotzdem vermieden werden.

Bei der Interaktion mit der Anwendung ist es weiterhin wichtig, dass der Benutzer nur in eine Richtung scrollen muss. BLok wurde daher auf die Breite der mobilen Geräte angepasst, sodass ein horizontales Scrollen entfällt. Der Benutzer muss aufgrund des umfangreichen Inhaltes von BLok lediglich in gewohnter Weise vertikal scrollen.

6.3.2.5 Formulare

Die Eingabe von Texten ist aufgrund der kleinen Tastatur bei mobilen Geräten für den Benutzer oft schwierig. Daher wurde bei BLok darauf geachtet, die notwendigen Eingaben auf ein Minimum zu reduzieren. Größtenteils werden daher Auswahllisten mit vorgegebenen Optionen genutzt. Die einzigen freien Eingaben sind zum Beispiel die einzelnen Einträge im Berichtsheft sowie die Kommentare und Bemerkungen.

Das Eingabefeld für die Dauer der Anwesenheit wurde mit Hilfe von HTML5 als Time Picker implementiert. Auf diese Weise wird zum Beispiel unter iOS der systemeigene Dialog zur Auswahl der Zeit genutzt. Der Benutzer muss dadurch die Zeit nicht mehr von Hand eingeben, sondern kann diese bequem über den Dialog einstellen. In Abbildung 95 ist diese Funktion dargestellt.

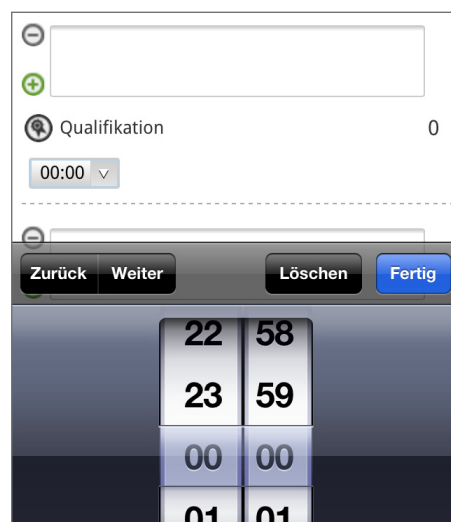


Abbildung 95 BLok: Auswahl der Zeit im Berichtsheft

Um das Ausfüllen von Formularen auch auf mobilen Endgeräten weiterhin zu optimieren, werden die Beschriftungen der Formularfelder stets über diesen angeordnet. Dadurch steht

mehr Platz für das Eingabefeld an sich zur Verfügung und auch beim Vergrößern der Ansicht bleibt die Beschriftung sichtbar.

6.3.3 Zugänglichkeit der Anwendung BLok

Für BLok wurde eine Webanwendung umgesetzt, sodass die Zugänglichkeit mit Hilfe von validem und semantischem HTML-Code sowie der Technologie WAI-ARIA sichergestellt wird. Da es sich aktuell bei BLok nur um einen Prototypen handelt, kann nur die Benutzeroberfläche auf die Zugänglichkeit untersucht werden. Ein weiterer Test sollte nach der Umsetzung der Funktionalitäten stattfinden, um die Anwendung erneut zu kontrollieren. An dieser Stelle ist auch ein Test mit Benutzern eines Screenreaders sinnvoll.

Die Kontrolle der Zugänglichkeit des Prototyps der Anwendung BLok findet mit dem Screenreader VoiceOver unter iOS statt.

Bei der Umsetzung der Oberfläche von BLok wurde besonders darauf geachtet, dass für alle Elemente Textalternativen hinterlegt sind. Dies trifft zum Beispiel auf Icons zu, die keinen zusätzlichen Beschreibungstext enthalten, wie das Plus und das Minus um Einträge hinzuzufügen oder zu entfernen. Auf diese Weise werden die Funktionen nicht nur für sehende Nutzer zugänglich gemacht, sondern auch für die Nutzer von Screenreadern. Diesen wird entsprechend „Neuen Eintrag hinzufügen“ bzw. „Eintrag entfernen“ vorgelesen, wie auch in Abbildung 96 zu sehen.

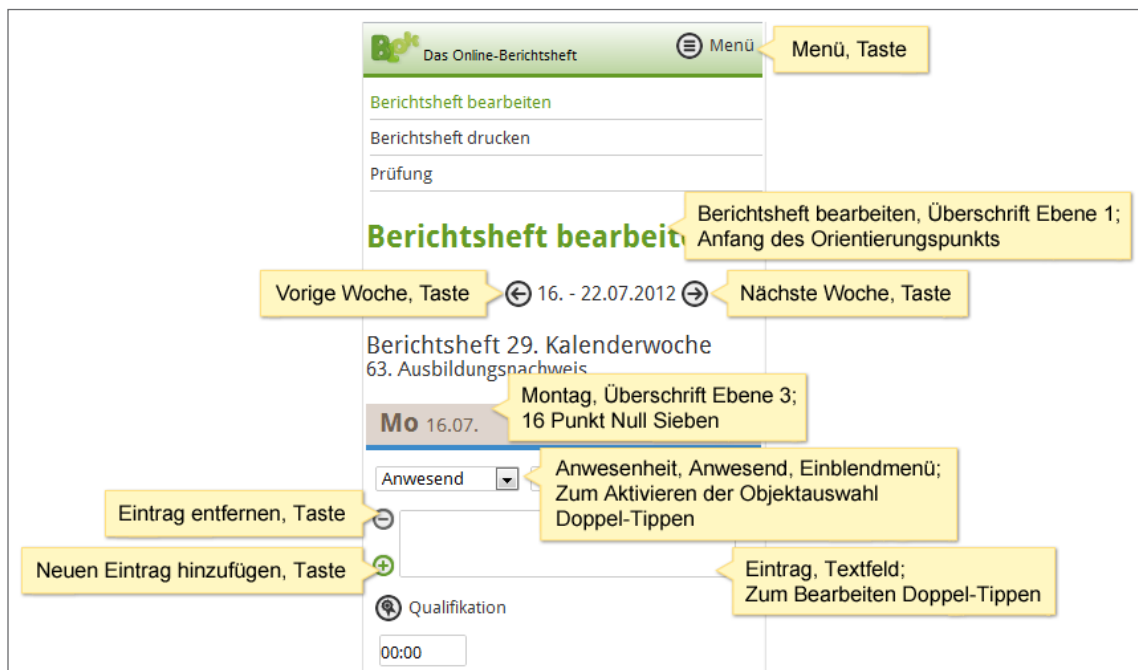


Abbildung 96 BLok: Verwendung von VoiceOver unter iOS

Grundsätzlich gehört zu jedem Formularelement auch ein Label. Im Berichtsheft wurden einige dieser allerdings weggelassen, um Platz zu sparen und die Übersichtlichkeit sicherzustellen. Dies betrifft zum Beispiel den Eintrag an sich oder die Auswahllisten für die Anwesenheit und den Aufenthaltsort. Für sehende Benutzer sind diese Elemente selbstbeschreibungsfähig, sodass nicht extra noch eine Beschriftung notwendig ist. Für Nutzer

von assistiven Technologien ist die zusätzliche Erklärung durch das Label aber durchaus notwendig, um den Zweck des Formularelementes zu verstehen. Daher werden diese für die sehenden Nutzer ausgeblendet, Screenreader können die Beschriftung aber vorlesen.

Die Ausgabe des Datums und der Zeit durch assistive Technologien ist noch problematisch, da diese nicht entsprechend vorgelesen werden. VoiceOver gibt zum Beispiel das Datum nur als Zahlen hintereinander aus. Dadurch wird das Verständnis des Inhaltes für den Benutzer erschwert. Datums- und Zeitangaben sollten grundsätzlich als solche sowohl zu erkennen sein als auch durch assistive Technologien ausgegeben werden können.

Ein weiteres Problem wird bei der Navigation deutlich. Beim Aufruf einer neuen Seite sollte der Screenreader oben auf der Seite mit dem Vorlesen beginnen. Ideal wäre das Ausgeben der Hauptüberschrift des Inhaltsbereiches, sodass der Benutzer sich orientieren kann. Aktuell beginnt die Ausgabe von VoiceOver aber weiter unten im Inhalt, sodass eine Orientierung schwer fällt.

6.3.4 Zusammenfassung BLok

Da für die Anwendung BLok zunächst ein Prototyp mit HTML und CSS erstellt wurde, konnte nur die Benutzeroberfläche bewertet werden. Die meisten Funktionalitäten sind noch nicht umgesetzt, sodass ein erneuter Test nach Fertigstellung der Anwendung sinnvoll ist. Es wird empfohlen weitere Benutzer in diese Analyse einzubeziehen, um eine objektive Sichtweise zu erhalten und BLok weiter zu verbessern.

Besonders wichtig wird ein Test mit realen Benutzern im Rahmen der Zugänglichkeit von BLok. Der Prototyp bildet bereits eine wichtige Grundlage zur Umsetzung der Accessibility durch die Verwendung der Technologie ARIA. Weitere Verbesserungen können aber nur in Zusammenarbeit mit Benutzern vorgenommen werden.

Grundsätzlich wirkt die Oberfläche von BLok aufgeräumter und übersichtlicher. Die einzelnen Bereiche der Anwendung sind klar strukturiert und voneinander abgegrenzt. Der Benutzer kann sich schnell orientieren und innerhalb von BLok zurechtfinden. Besonders die Trennung der Pflichtangaben von den optionalen Funktionen im Berichtsheft helfen dem Benutzer bei der Konzentration auf die Arbeitsaufgabe. Er wird nicht von Optionen abgelenkt, die er gerade nicht benötigt.

Durch das Verwenden der Technologie Responsive Webdesign bei der Umsetzung von BLok kann die Anwendung sowohl auf mobilen Geräten mit kleinen Bildschirmen als auch auf einem Desktop-Rechner genutzt werden. Das Layout ist flexibel gehalten, sodass es sich an verschiedene Bildschirmgrößen anpasst.

7 Zusammenfassung und Ausblick

Das Ziel der vorliegenden Masterarbeit war die Konzeption und die prototypische Umsetzung der Anwendung BLok - Das Online Berichtsheft. BLok sollte nicht nur für die Verwendung mit einem herkömmlichen Desktop-Rechner optimiert werden, sondern auch für mobile Geräte mit Touchscreen. Um diese Anforderung zu erfüllen war eine Recherche zu den Themen Usability und Accessibility notwendig. Im Rahmen der theoretischen Grundlagen wurden somit wichtige Richtlinien und Konzepte allgemein zu Usability und Accessibility sowie zu mobilen Anwendungen und nativen Apps zusammengetragen.

Aufgrund der abweichenden Größe und Bedienung sowie der technischen Möglichkeiten der mobilen Endgeräte war zusätzlich eine Recherche zu modernen Webtechnologien notwendig, die diese Eigenschaften berücksichtigen. Neben Technologien wie HTML5 und Responsive Webdesign zur Umsetzung einer Anwendung wurden auch Methoden berücksichtigt, welche die Zugänglichkeit von Anwendungen verbessern. Dabei handelt es sich zum Beispiel um ARIA oder Maßnahmen zur Optimierung der Zugänglichkeit unter iOS und Android.

Neben den theoretischen Grundlagen wurde außerdem eine Analyse der Anwendungen Evernote und Google Mail durchgeführt. Als Basis dieser Analyse dienten die allgemeinen sowie die speziellen Kriterien und Richtlinien der Usability und Accessibility für den mobilen Kontext. Durch die Analyse konnte ein Einblick in die praktische Umsetzung der theoretischen Richtlinien und Empfehlungen gewonnen werden. Positive Ansätze der Umsetzung wurden somit in das Konzept von BLok eingearbeitet.

Nach der Recherche der notwendigen theoretischen Grundlagen sowie Webtechnologien und der Analyse der Anwendungen Evernote und Google Mail wurde ein Konzept für BLok erarbeitet. Dieses umfasst sowohl den strukturellen Aufbau als auch ein Bedienkonzept für die mobilen Geräte mit Touchscreen. Entsprechend der Vorgaben entstanden grafische Entwürfe für die verschiedenen Ansichten der Anwendung.

Für die technische Umsetzung von BLok wurde sich zunächst gegen eine native App und stattdessen für eine Webanwendung entschieden. Gründe hierfür waren eine einfache und Ressourcen sparende Realisierung, eine bessere Verbreitung durch das World Wide Web sowie die leichte Wartbarkeit. Weiterhin wurde eine Webanwendung für alle Geräte umgesetzt, sodass keine separate Version nur für den mobilen Kontext entstand. Die Entscheidung für eine Webanwendung, die sich flexibel an alle Bildschirmgrößen anpassen soll, bedingt die Auswahl der Technologie Responsive Webdesign. Mit Hilfe dieser Technologie entstand ein flexibles Layout, welches sich dynamisch an verschiedene Bildschirmgrößen anpasst. Diese Flexibilität wurde mit Hilfe der CSS3 Media Queries realisiert.

Als Ergebnis der Masterarbeit liegt ein Prototyp der Webanwendung BLok vor. Dieser wurde mit Hilfe der Technologien HTML und CSS sowie ARIA zur Unterstützung der Zugänglichkeit umgesetzt. Bei einer Veränderung der Bildschirmgröße passt sich das Layout flexibel an, sodass BLok sowohl mit einem Desktop-Computer als auch mit einem mobilen Gerät wie zum Beispiel dem iPhone genutzt werden kann. Dabei ist die Anwendung durch die Verwendung moderner Technologien und Konzepte wie HTML5 und Responsive Webdesign zukunftssicher gestaltet. Für neu erscheinende Geräte mit anderen Bildschirmgrößen oder Auflösungen müssen zum Beispiel keine Anpassungen vorgenommen werden. Trotzdem

bleibt die Anwendung BLok abwärtskompatibel, sodass auch Benutzer älterer Technologien diese verwenden können.

Auf Grundlage des entstandenen Prototypen sollten nun die Funktionalitäten umgesetzt werden. Diese umfassen nicht nur allgemeine Funktionen wie die Verarbeitung und Speicherung sowie Ausgabe von Daten, sondern auch Anpassungen für die mobilen Geräte. Dem Benutzer sollten die technischen Möglichkeiten, wie zum Beispiel die Ortung mittels GPS und die Verwendung der Anwendung ohne Verbindung zum Internet auch in BLok zur Verfügung gestellt werden. Durch die vorhandenen Spezifikationen Geolocation API und Offline Web Applications ist die Implementierung auch im Rahmen einer Webanwendung einfach umsetzbar. Weiterhin sollte es dem Benutzer möglich sein, Dateien hinzuzufügen und mit der Kamera des mobilen Gerätes direkt aus der Anwendung heraus Fotos aufzunehmen und diese in BLok einzubinden.

Ist die grundlegende funktionale Umsetzung abgeschlossen, sollten so früh wie möglich reale Benutzer in die Evaluation der entstandenen Anwendung einbezogen werden. Auf diese Weise wird nicht nur die korrekte Funktionsweise sichergestellt, sondern auch die Usability evaluiert. Durch den Nutzertest können weitere Verbesserungen identifiziert und umgesetzt werden. Zu beachten ist bei diesen Tests, dass sowohl unterschiedliche Geräte als auch verschiedene Browser genutzt werden sollten. Mögliche Schwierigkeiten können auf diesem Wege identifiziert und behoben werden.

In einem weiteren Test sollte die Zugänglichkeit der Anwendung untersucht werden. Für BLok stehen zunächst vor allem Benutzer mit kognitiven Behinderungen im Vordergrund. Um diese besser bei der Nutzung zu unterstützen wurden Symbole und Texte kombiniert sowie auf leicht verständliche Formulierungen geachtet. Die Benutzer sollten dies nun testen und bewerten, sodass gegebenenfalls weitere Verbesserungen vorgenommen werden können. Im Zusammenhang mit der Zugänglichkeit von BLok sollte außerdem evaluiert werden, ob auch Nutzer mit körperlichen Behinderungen die Anwendung verstärkt nutzen wollen. In diesem Fall liegt mit dem Prototyp von BLok bereits eine gute Grundlage vor, da auch bei der technischen Umsetzung auf die Accessibility geachtet wurde. Durch das Einbeziehen betroffener Nutzer kann die Anwendung in diesem Bereich weiter optimiert werden.

Auch eine Befragung zum Thema mobile Nutzung bietet sich im Rahmen von BLok an. Auf diese Weise könnte festgestellt werden, wie intensiv die Benutzer von BLok mobile Geräte einsetzen, um ihr Berichtsheft zu pflegen. Ist die Nachfrage entsprechend groß kann das Anbieten nativer Apps in Betracht gezogen werden. Durch erweiterte funktionale Möglichkeiten und eine bessere Usability durch die Nutzung der Komponenten des jeweiligen Betriebssystems, kann die Zufriedenheit der Benutzer noch gesteigert werden. Außerdem ergeben sich durch die Verwendung von nativen Apps neue Vertriebsmöglichkeiten durch das Verkaufen der App über die App Stores der verschiedenen Hersteller.

Aus den noch bevorstehenden Arbeitsschritten bei der Umsetzung der Anwendung BLok ergibt sich ein Plan für die Veröffentlichung. Dieser Release-Plan soll Schritte und Termine definieren, welche für die pünktliche und vollständige Fertigstellung notwendig sind. Die Veröffentlichung von BLok und die Bereitstellung für alle Kunden ist für Mitte Januar 2013 festgelegt. Es ist daher notwendig, die Umsetzung im Oktober 2012 abzuschließen, sodass die korrekte Funktionsweise langfristig getestet werden kann. Diese Kontrolle findet im Rahmen einer internen und externen Qualitätssicherung statt. Die interne Qualitätssicherung

mit zahlreichen manuellen und automatischen Tests sowie einer fortlaufenden Korrektur auftauchender Probleme ist für November 2012 geplant. Im Rahmen der externen Qualitätssicherung soll ausgewählten Kunden ein Demo-System bereitgestellt werden. Auf diese Weise können die Benutzer das System ausführlich mit ihren jeweiligen realistischen Szenarien und Arbeitsabläufen testen. Gemeldete Probleme oder Verbesserungen sollten in dieser Phase, die im Dezember 2012 stattfindet, aufgenommen und umgesetzt werden.

Im Rahmen der Nachbereitung im Anschluss an die Veröffentlichung im Januar 2013 ist nochmals eine Kontrolle der Funktionsfähigkeit der Anwendung zu empfehlen. Auch nach der Veröffentlichung wird es Empfehlungen durch die Benutzer geben, sodass eine fortlaufende Verbesserung von BLok stattfindet.

Weiterhin können Befragungen der Benutzer zur mobilen Version von BLok vorgenommen werden. Ist die Resonanz positiv und eine vermehrte Nutzung der Anwendung mit mobilen Geräten nachvollziehbar, sollte das Anbieten von nativen Apps erneut in Betracht gezogen werden.

Anlagen

Verzeichnis der Anlagen

Anlage A	Bewertung der Anwendungen Evernote und Google Mail.....	155
Anlage B	Konzept für die Webanwendung BLok	157
Anlage C	BLok Styleguide	161
Anlage D	Bewertung der Anwendung BLok.....	165

Anlage A Bewertung der Anwendungen Evernote und Google Mail

Kriterium	Bewertung Evernote	Bewertung Google Mail
ISO 9241-110: Dialoggestaltung		
Aufgabenangemessenheit	1	2
Selbstbeschreibungsfähigkeit	2	1
Erwartungskonformität	1	1
Lernförderlichkeit	1	2
Steuerbarkeit	1	1
Fehlertoleranz	2	3
Individualisierbarkeit	2	3
Mobile Usability		
Konsistenz	2	2
Navigation	2	1
Informationsdarstellung	1	2
Interaktionen	2	3
Formulare	1	2
Zugänglichkeit		
	2	4
Gesamtbewertung		
	1,5	2

Anlage B Konzept für die Webanwendung BLok

Entwurf für die Desktop-Variante

Blok - Das Online Berichtsheft

Abmelden

Übersicht

Berichtsheft

Menu Item

Menu Item

Nutzername

Berichtsheft bearbeiten

Berichtsheft drucken

Prüfung

Berichtsheft für die 29. Kalenderwoche

Ausbildungsnachweis Nr. 62

Heute

16. - 22.07.2012

Woche

Monat

Jahr

MO

16. JULY

Anwesend

Betrieb

[-]

[-]

[+]

Qualifikation

2

0h:00min

Qualifikation

0

0h:00min

Gesamtstunden:

0h:00min

Datei anhängen

Kommentar

Gespeichert

DI

17. JULY

Anwesend

Betrieb

[-]

[-]

[+]

Qualifikation

0

0h:00min

Gesamtstunden:

0h:00min

Datei anhängen

Kommentar

Gesamtstunden:

0h:00min

Bemerkungen für diese Woche hinzufügen:

Berichtsheft freigeben

Folgende Ausbilder über die Freigabe benachrichtigen:

Bitte wählen

Für die Zeit im Betrieb.

Freigeben

Entwurf für die Tablet-Variante

Blok - Das Online Berichtsheft

Abmelden

ÜbersichtBerichtsheftMenu ItemMenu Item

Nutzername

Berichtsheft bearbeiten | Berichtsheft drucken | Prüfung

Berichtsheft für die 29. Kalenderwoche
Ausbildungsnachweis Nr. 62

Heute◀16. - 22.07.2012▶WocheMonatJahr

Montag, 16.7.2012

AnwesendBetrieb

—

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex.

Qualifikationen24h:00min

—

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore

Qualifikationen00h:00min

+ Neuer Eintrag

Gesamtzeit: 4h:00min

Datei anhängen

Kommentar

✓ Gespeichert

Dienstag, 17.7.2012

AnwesendBetrieb

—


Qualifikationen00h:00min

+ Neuer Eintrag

Entwurf für die Smartphone-Variante

Startseite

BLOK - Das Online Berichtsheft Menü



Willkommen bei BLOK!
Nutzername

Neuigkeiten

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam

Nachrichten

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam

Menü

Übersicht

Berichtsheft

Ausbilder

Nachrichten

Entwicklungsportfolio

Einstellungen

Abmelden

Ansicht Auszubildender

BLOK Berichtsheft Menü

Berichtsheft bearbeiten

Berichtsheft drucken

Prüfung

16. - 22.07.2012

▼ Montag, 16.7.2012

Anwesend Betrieb

—

Lorem ipsum dolor sit amet, consetetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex.

Qualifikationen

2

 4h:00min

—

Lorem ipsum dolor sit amet, consetetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore

Qualifikationen

0

 0h:00min

+

 Neuer Eintrag

1

2

Gesamtzeit: 4h:00min

► Dienstag, 17.7.2012

Krank

1

► Mittwoch, 18.7.2012

Betrieb, 0h:00min

► Donnerstag, 19.7.2012

► Freitag, 20.7.2012

Gesamtzeit der Woche: 4h:00min

Bemerkungen für diese Woche hinzufügen:

Berichtsheft freigeben
Folgende Ausbilder über die Freigabe benachrichtigen:

Bitte wählen Für die Zeit im Betrieb.

Bitte wählen Für die Zeit in der Schule.

Freigeben

Menü

Übersicht

Berichtsheft

Ausbilder

Nachrichten

Entwicklungsportfolio

Einstellungen

Ansicht Ausbilder

BLOK Berichtsheft

Überblick Azubi

Berichtsheft einsehen

16. - 22.07.2012

Ausgewählter Azubi

Name Azubi

!

 Hinweis: Diese Woche befindet sich noch in Bearbeitung

▼ Montag, 16.7.2012

Anwesend Betrieb

—

Lorem ipsum dolor sit amet, consetetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex.

2

 Qualifikation 1 4h:00min

2

 Qualifikation 2 [Weitere anzeigen](#)

—

Lorem ipsum dolor sit amet, consetetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore

0

 0h:00min

Dateien

[test.pdf](#)

2

[test.pdf](#)

Gesamtzeit: 4h:00min

► Dienstag, 17.7.2012

Krank

1

► Mittwoch, 18.7.2012

Betrieb, 0h:00min

► Donnerstag, 19.7.2012

► Freitag, 20.7.2012

Gesamtzeit der Woche: 4h:00min

Bemerkungen für diese Woche hinzufügen:

Abnahme des Berichtshefts

Akzeptieren

Ablehnen

Menü

Übersicht

Berichtsheft

Ausbilder

Nachrichten

Entwicklungsportfolio

Einstellungen

Anlage C BLok Styleguide



BLok Styleguide - Web und Mobil

BPS Bildungsportal Sachsen GmbH
August 2012

Logo & Farbwahl

BLok Logo - normal



BLok Logo - reduziert



BLok Logo - reduziert, negativ



Primäre Farben

#D7DF73	#B7CF3F	#649B26
#DED4CD	#ADA194	#706152
#C8C8C8	#646464	#323232

Sekundäre Farben

#F7EBD1	#FAD6A3	#E2AF38
#E28690	#BF0426	#690011
#B8D2E8	#3A8BCE	#065991

Farben für die Benutzerrollen

	Azubi R: 99, G: 156, B: 38 #649B26
	Leiter der Ausbildung R: 191, G: 4, B: 38 #BF0426
	Betrieblicher Ausbilder R: 59, G: 140, B: 207 #3A8BCE
	Externer Ausbilder R: 112, G: 97, B: 82 #706152
	Schulischer Ausbilder R: 226, G: 175, B: 56 #E2AF38

Typografie

Schriftart: Droid Sans

Regular

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet.

Bold

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet.

Schriftgrad

Ausgangswert: 98%

Fließtext: 1em

Sekundärer Inhalt: 0.85em

Überschrift Ebene 1

Droid Sans, Bold, 1.5em, #649B26

Überschrift Ebene 2

Droid Sans, 1.2em, #323232

Überschrift Ebene 3

Droid Sans, Bold, 1.0em, #323232

Überschrift Ebene 1

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet.

Überschrift Ebene 2

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet.

Überschrift Ebene 3

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At

Gestaltung von sekundärem Inhalt

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At

Interaktionselemente

Link normal

Links werden für den Aufruf von internen oder externen Seiten genutzt

Link normal | [hover](#) | [focus](#) | #649B26

Button mit Icon

Um wichtige Funktionen hervorzuheben werden Links mit Icons verwendet. Die Aufmerksamkeit des Benutzers soll mit diesen Interaktionselementen gelenkt werden. Wichtige Funktionen sind zum Beispiel Registrieren und Abmelden.

Durch den Einsatz einer anderen Farbe kann die Wichtigkeit der Funktion stärker hervorgehoben werden.

 Noch nicht registriert?

 Datei anhängen

 Kommentar hinzufügen

Abmelden 

Button normal

Buttons werden für das Auslösen von Funktionen, wie zum Beispiel Speichern, Abbrechen, Freigeben usw. verwendet

Zustände: normal | hover | hervorgehoben für noch nicht gespeicherte Änderungen

Meldungen

Hinweis

Das ist ein Hinweis

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy

Warnung / Achtung

Das ist eine Warnung

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy

Fehler

Das ist ein Fehler

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy

Icons & Grafiken

Haupt-Navigation

Zustand normal



Zustand mouseover



Zustand aktiv



Zustand nicht verfügbar



Gestaltung der Startseite

Informationen, News und Vollständigkeit



Funktionen



Kennzeichnung von bereits vorhandenen und neuen Elementen

Für die Anzeige vorhandener oder neuer Dateien, Kommentare oder Nachrichten.



Wichtige Funktionen



Grafiken Benutzerrollen



Auszubildender



Leiter der Ausbildung



Betrieblicher



Externer



Schulischer Ausbilder

Gestaltung der Webanwendung

Header mit Navigation



Footer



Login

Beim Login wird der Header weg gelassen, da weder Navigation noch Profilbild notwendig sind. Stattdessen wird nur das Blok-Logo angezeigt.



Das Online-Berichtsheft

Willkommen bei Blok

Anmelden

Benutzername

Passwort

Benutzername vergessen? Passwort vergessen?

☐ Noch nicht registriert?

Neuigkeiten

NEWS NEWS NEWS

Startseite

Willkommen bei Blok

Informationen

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat, sed diam voluptua. Ut vero temp et accusam et justo duo dolores et ea rebum. Sed utra sed gubergin, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur

Vollständigkeit

Lorem ipsum dolor sit amet, consetetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat, sed diam voluptua. Ut vero temp et accusam et justo duo dolores et ea rebum. Sed utra sed gubergin, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur

Neuigkeiten

Lorem ipsum dolor sit amet, consetetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat, sed diam voluptua. Ut vero temp et accusam et justo duo dolores et ea rebum. Sed utra sed gubergin, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur

Berichtsheft

Der aktuelle Tag wird mit einem hellen Braun und dunkler Schriftfarbe ausgezeichnet. Weiterhin wird bei jedem Tag unterhalb der Box eine farbige Linie gesetzt, um den aktuellen Aufenthaltsort des Auszubildenden zu kennzeichnen. Die Farbe entspricht dabei der Farbe der Rolle des jeweiligen Ausbilders.

Berichtsheft 29. Kalenderwoche

63. Ausbildungsnachweis

Mein 18. - 22.07.2012 Ansicht: Woche Jahr

Mo 18.07. Di 19.07. Mi 20.07.

☐ Eintrag ☐ Eintrag

☐ Eintrag ☐ Eintrag

☒ Eintrag hinzufügen

Gesamtstunden Tag: 0h:00min

☒ Gruppensicht

☒ Detail anhängen ☒ Kommentieren

Besonderheiten der mobilen Anwendung

Gestaltung des Kopfbereichs

Startseite: Logo der Anwendung dominanter, Button Menü zum Aufruf der Navigation



Inhalt: Logo der Anwendung reduziert, Titel der aktuellen Seite direkt im Header platziert



Gestaltung der Navigation

Die Haupt-Navigation wird mit dem Button Menü im Header aufgerufen.
Die Navigation ist in Form einer Liste gestaltet, die über die gesamte Breite des Bildschirms geht.



Gestaltung des Berichtshefts

Die Übersicht über das Berichtsheft wird ebenfalls in Form einer Liste gestaltet. In der Liste befinden sich Informationen zum aktuellen Tag, zu dessen Status und Hinweise auf neue Dateien oder Kommentare.
Per Tap kann ein Tag aufgeklappt und bearbeitet werden. Der geöffnete Tag wird farblich hervorgehoben.



Besonderheiten der iOS App

Gestaltung des Kopfbereichs

Startseite: Logo der Anwendung als Hauptüberschrift, Button Einstellungen



Inhalt: Titel der aktuellen Ansicht direkt im Header platziert, Zurück Button auf der linken Seite



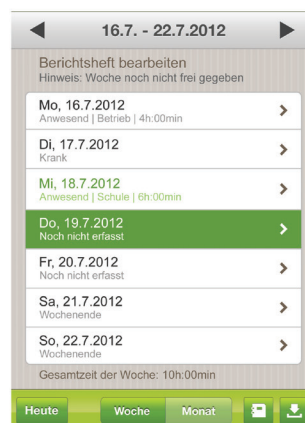
Gestaltung der Navigation

Die Navigation ist in Form einer Liste gestaltet. Ein per Tap ausgewähltes Element wird grün hervorgehoben, bis sich die nächste Ansicht öffnet.
Neue Nachrichten können über ein zusätzliches Symbol dargestellt werden.



Gestaltung des Berichtshefts

Die Übersicht über das Berichtsheft wird in Form einer Liste gestaltet. In der Liste befinden sich Informationen zum aktuellen Tag und zu dessen Status. Wählt der Benutzer einen Tag aus, kommt er in die Detailsansicht und kann die zugehörigen Einträge bearbeiten.



Grundsätzlich sind iOS Standardelemente zur Umsetzung zu verwenden.

Anlage D Bewertung der Anwendung BLok

Kriterium	Bewertung BLok
ISO 9241-110: Dialoggestaltung	
Aufgabenangemessenheit	1
Selbstbeschreibungsfähigkeit	2
Erwartungskonformität	1
Lernförderlichkeit	2
Steuerbarkeit	2
Fehlertoleranz	2
Individualisierbarkeit	1
Mobile Usability	
Konsistenz	1
Navigation	3
Informationsdarstellung	2
Interaktionen	2
Formulare	1
Zugänglichkeit	
	2
Gesamtbewertung	
	1,7

Literaturverzeichnis

A

Adapt.js: Adaptive CSS.

URL: <<http://adapt.960.gs/>>, verfügbar am 25.6.2012

Adobe Systems Inc.: PhoneGap, About the Project.

URL: <<http://phonegap.com/about>>, verfügbar am 6.8.2012

Adobe Systems Inc.: PhoneGap, PhoneGap Explained Visually, 2.5.2012.

URL: <<http://phonegap.com/2012/05/02/phonegap-explained-visually/>>, verfügbar am 6.8.2012

Android Developer: Android 4.0, Ice Cream Sandwich.

URL: <<http://developer.android.com/sdk/android-4.0-highlights.html>>, verfügbar am 7.6.2012

Android Developer: Android Design, Action Bar.

URL: <<http://developer.android.com/design/patterns/actionbar.html>>, verfügbar am 12.4.2012

Android Developer: Android Design, Design Principles.

URL: <<http://developer.android.com/design/get-started/principles.html>>, verfügbar am 12.4.2012

Android Developer: Android Design, Gestures.

URL: <<http://developer.android.com/design/patterns/gestures.html>>, verfügbar am 12.4.2012

Android Developer: Android Design, Grid Lists.

URL: <<http://developer.android.com/design/building-blocks/grid-lists.html>>, verfügbar am 12.4.2012

Android Developer: Android Design, Iconography.

URL: <<http://developer.android.com/design/style/iconography.html>>, verfügbar am 12.4.2012

Android Developer: Android Design, Lists.

URL: <<http://developer.android.com/design/building-blocks/lists.html>>, verfügbar am 12.4.2012

Android Developer: Android Design, Navigation.

URL: <<http://developer.android.com/design/patterns/navigation.html>>, verfügbar am 12.4.2012

Android Developer: Android Design, Pure Android.

URL: <<http://developer.android.com/design/patterns/pure-android.html>>, verfügbar am 12.4.2012

Android Developer: Android Design, Spinners.

URL: <<http://developer.android.com/design/building-blocks/spinners.html>>, verfügbar am 12.4.2012

Android Developer: Android Design, Text Fields.

URL: <<http://developer.android.com/design/building-blocks/text-fields.html>>, verfügbar am 12.4.2012

Android Developer: Android Design, Themes.

URL: <<http://developer.android.com/design/style/themes.html>>, verfügbar am 9.7.2012

Android Developer: Android Design, UI Overview.

URL: <<http://developer.android.com/design/get-started/ui-overview.html>>, verfügbar am 12.4.2012

Android Developer: Develop, Android Training.

URL: <<http://developer.android.com/training/index.html>>, verfügbar am 18.6.2012

Android Developer: Develop, Making Applications Accessible.

URL: <<http://developer.android.com/guide/topics/ui/accessibility/apps.html>>, verfügbar am 18.4.2012

Apple: Accessibility, iPhone.

URL: <<http://www.apple.com/accessibility/iphone/vision.html>>, verfügbar am 18.7.2012

Apple: Accessibility Programming Guide for iOS, Accessibility on iPhone.

URL: <https://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/iPhoneAccessibility/Accessibility_on_iPhone/Accessibility_on_iPhone.html>, verfügbar am 18.4.2012

Apple: Accessibility Programming Guide for iOS, Making Your iPhone Application Accessible.

URL: <https://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/iPhoneAccessibility/Making_Application_Accessible/Making_Application_Accessible.html>, verfügbar am 18.4.2012

Apple Developer: Developer Tools.

URL: <<https://developer.apple.com/technologies/tools/>>, verfügbar am 27.6.2012

Apple: iOS Human Interface Guidelines, App Design Strategies.

URL: <<http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/AppDesign/AppDesign.html>>, verfügbar am 5.4.2012

Apple: iOS Human Interface Guidelines, Case Studies: Transitioning to iOS.

URL: <<http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/TranslateApp/TranslateApp.html>>, verfügbar am 9.7.2012

Apple: iOS Human Interface Guidelines, Custom Icon and Image Creation Guidelines.

URL: <<http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/IconImages/IconImages.html>>, verfügbar am 5.4.2012

Apple: iOS Human Interface Guidelines, Human Interface Principles.

URL: <<http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/Principles/Principles.html>>, verfügbar am 5.4.2012

Apple: iOS Human Interface Guidelines, iOS UI Element Usage Guidelines.

URL: <<http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/UIElementGuidelines/UIElementGuidelines.html>>, verfügbar am 5.4.2012

Apple: iOS Human Interface Guidelines, Platform Characteristics.

URL: <<http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/Characteristics/Characteristics.html>>, verfügbar am 5.4.2012

Apple: iOS Human Interface Guidelines, User Experience Guidelines.

URL: <<http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/UEBestPractices/UEBestPractices.html>>, verfügbar am 5.4.2012

Apple: iPhone, iOS.

URL: <<http://www.apple.com/de/iphone/ios/>>, verfügbar am 7.6.2012

B

Böhringer, Joachim ; Bühler, Peter ; Schlaich, Patrick: Kompendium der Mediengestaltung für Digital- und Printmedien : Konzeption und Gestaltung für Digital- und Printmedien. - 4. Auflage - Berlin : Springer, 2008

Bundesministerium der Justiz: Gesetz zur Gleichstellung behinderter Menschen, § 4 Barrierefreiheit.

URL: <<http://www.gesetze-im-internet.de/bgg/BJNR146800002.html>>, verfügbar am 18.4.2012

C

Clark, Josh: Nielsen is wrong on mobile, 12.4.2012.

URL: <<http://www.netmagazine.com/opinions/nielsen-wrong-mobile>>, verfügbar am 16.4.2012

Combrinck, Tanya: Designers respond to Nielsen on mobile, 12.4.2012.

URL: <<http://www.netmagazine.com/news/designers-respond-nielsen-mobile-121892>>, verfügbar am 16.4.2012

Croft, Jeff: How I'm implementing Responsive Web Design, 3.2.2012.

URL: <<http://jeffcroft.com/blog/2012/mar/02/implementing-responsive-design/>>, verfügbar am 19.3.2012

CSS Tricks: Media Queries for Standard Devices.

URL: <<http://css-tricks.com/snippets/css/media-queries-for-standard-devices/>>, verfügbar am 25.6.2012

D

Dahm, Markus: Grundlagen der Mensch-Computer-Interaktion. - München : Pearson Studium, 2006

Deveria, Alexis: When can i use, Offline Web Applications.

URL: <<http://caniuse.com/#feat=offline-apps>>, verfügbar am 3.8.2012

DIN EN ISO 9241 Teil 11:1998. Anforderungen an die Gebrauchstauglichkeit; Leitsätze

DIN EN ISO 9241 Teil 12:1998. Informationsdarstellung

DIN EN ISO 9241 Teil 110:2006. Grundsätze der Dialoggestaltung

E

Evernote Corporation: Evernote, Download.

URL: <<http://evernote.com/intl/de/evernote/index.php>>, verfügbar am 14.5.2012

Evernote Corporation: Evernote, Ihr virtuelles Gedächtnis.

URL: <<http://evernote.com/intl/de/>>, verfügbar am 14.5.2012

F

Featherstone, Derek: Real World Accessibility, HTML5, ARIA and the Modern Web, 5.7.2011.

URL: <<http://www.sitepoint.com/real-world-accessibility-html5-aria-and-the-modern-web>>, verfügbar am 20.4.2012

Firtman, Maximiliano: jQuery Mobile : Up and Running. - 1. Auflage - Sebastopol : O'Reilly, 2012

Firtman, Maximiliano: Programming the Mobile Web : Reaching Users on iPhone, Android, BlackBerry, Symbian, and more. - 1. Auflage - Sebastopol : O'Reilly, 2010

Fling, Brian: Mobile Design and Development : Practical Techniques for Creating Mobile Sites and Web Apps. - 1. Auflage - Sebastopol : O'Reilly, 2009

G

Gibson, Alex: Making web content more accessible with iOS VoiceOver, 6.6.2011.
URL: <<http://alxgbsn.co.uk/2011/06/06/making-web-content-more-accessible-with-ios-voiceover/>>, verfügbar am 18.7.2012

Google Mobile: Google Mail für Handys.
URL: <<http://www.google.com/intl/de/mobile/mail/>>, verfügbar am 3.7.2012

Gustafson, Aaron: Understanding Progressive Enhancement, 7.10.2008.
URL: <<http://www.alistapart.com/articles/understandingprogressiveenhancement/>>, verfügbar am 21.5.2012

H

Hadlock, Kris: jQuery Mobile : Develop and Design. - Berkeley : Pearson Education, 2012

Handbuch Usability: Definition Usability.
URL: <<http://www.handbuch-usability.de/begriffsdefinition.html>>, verfügbar am 2.5.2012

Herczeg, Michael: Software-Ergonomie : Theorien, Modelle und Kriterien für gebrauchstaugliche interaktive Computersysteme. - 3. Auflage - München : Oldenbourg Wissenschaftsverlag, 2009

Hogan, Brian P.: HTML5 & CSS3 : Webentwicklung mit den Standards von morgen. - 1. Auflage - Köln : O'Reilly, 2011

Hoober, Steven ; Berkman, Eric: Designing Mobile Interfaces, General Interactive Controls.
URL: <<http://4ourth.com/wiki/General%20Interactive%20Controls>>, verfügbar am 5.3.2012

Hoober, Steven ; Berkman, Eric: Designing Mobile Interfaces, Page.
URL: <<http://4ourth.com/wiki/Page>>, verfügbar am 5.3.2012

Hoober, Steven ; Berkman, Eric: Designing Mobile Interfaces : Patterns for Interaction Design. - 1. Auflage - Sebastopol : O'Reilly, 2011

Hoober, Steven ; Berkman, Eric: Designing Mobile Interfaces, Principles of Mobile Design.
URL: <<http://4ourth.com/wiki/Principles%20of%20Mobile%20Design>>, verfügbar am 5.3.2012

Hoober, Steven ; Berkman, Eric: Designing Mobile Interfaces, Scroll.
URL: <<http://4ourth.com/wiki/Scroll>>, verfügbar am 5.3.2012

J

Jankord, Brett: Hybrid Responsive Web Design, 29.2.2012.

URL: <<http://www.brettjankord.com/2012/02/29/hrwd-hybrid-responsive-web-design/>>, verfügbar am 29.5.2012

Johnson, Joshua: Mobile First Design, Why It's Great and Why It Sucks, 13.3.2012.

URL: <<http://designshack.net/articles/css/mobilefirst/>>, verfügbar am 1.4.2012

Johnson, Joshua: Responsive Design, Why You're Doing It Wrong, 22.2.2012.

URL: <<http://designshack.net/articles/css/responsive-design-why-youre-doing-it-wrong/>>, verfügbar am 1.4.2012

jQuery Mobile Framework: Download.

URL: <<http://jquerymobile.com/download/>>, verfügbar am 29.6.2012

K

Keck, Moritz: App-Usability, Herausforderungen und Guidelines, in: Forschungsbeiträge der eResult GmbH, 2010.

URL: <http://www.eresult.de/studien_artikel/forschungsbeitraege/app-usability.html>, verfügbar am 29.2.2012

Kröner, Peter: HTML5 : Webseiten innovativ und zukunftssicher. - 2. Auflage - München : Open Source Press, 2011

L

Lawson, Bruce ; Sharp, Remy: Introducing HTML5 : Second Edition. - 2. Auflage - Berkeley : New Riders Press, 2012

Layon, Kristofer: Mobilizing Web Sites : Develop and Design Strategies for Mobile Web Implementation. - 1.Auflage - Berkeley : Peachpit Press, 2012

M

Marcotte, Ethan: Responsive Web Design. - 1.Auflage - New York : A Book Apart, 2011

Microsoft: Designing Applications for Windows Mobile Platforms, Accessibility and Ergonomic Guidelines, 19.4.2012.

URL: <<http://msdn.microsoft.com/en-us/library/bb158589.aspx>>, verfügbar am 12.4.2012

Microsoft: User Experience Design Guidelines, Central Application Hub with Home Page Menu (Panorama or Pivot Control) for Windows Phone.

URL: <<http://msdn.microsoft.com/en-us/library/hh202892>>, verfügbar am 30.4.2012

Microsoft: User Experience Design Guidelines, General Design Principles.

URL: <<http://msdn.microsoft.com/en-us/library/hh202906>>, verfügbar am 30.4.2012

Microsoft: User Experience Design Guidelines, List with Details Drilldown for Windows Phone.

URL: <<http://msdn.microsoft.com/en-us/library/hh202886>>, verfügbar am 30.4.2012

Microsoft: User Experience Design Guidelines, Panorama Control Overview for Windows Phone.

URL: <<http://msdn.microsoft.com/en-us/library/ff941104>>, verfügbar am 30.4.2012

Microsoft: User Experience Design Guidelines, Pivot Control Design Guidelines for Windows Phone.

URL: <<http://msdn.microsoft.com/en-us/library/hh202919>>, verfügbar am 30.4.2012

Microsoft: User Experience Design Guidelines, TextBox Control Design Guidelines for Windows Phone.

URL: <<http://msdn.microsoft.com/en-us/library/hh202916>>, verfügbar am 30.4.2012

Microsoft: WindowsPhone, Schritt für Schritt zur eigenen App.

URL: <<http://msdn.microsoft.com/de-de/windowsphone/hh442446>>, verfügbar am 27.6.2012

Mobile Moxie: 16 Mobile Site Usability Mistakes to Avoid, 13.2.2012.

URL: <<http://www.mobilemoxie.com/2012/02/mobile-site-usability-mistakes/>>, verfügbar am 29.2.2012

N

Nielsen, Jakob: Mobile Site vs. Full Site, 10.4.2012.

URL: <<http://www.useit.com/alertbox/mobile-vs-full-sites.html>>, verfügbar am 13.4.2012

Nyman, Lewis: Techniques For Gracefully Degrading Media Queries, 10.8.2011.

URL: <<http://coding.smashingmagazine.com/2011/08/10/techniques-for-gracefully-degrading-media-queries/>>, verfügbar am 28.8.2012

S

Saffer, Dan: Designing Gestural Interfaces : Touchscreens and Interactive Devices. - 1. Auflage - Sebastopol : O'Reilly, 2008

Seward, Dan: Designing Usable Mobile Websites : A Practical Guide. - 1. Auflage - Brisbane : Peak Usability, 2011

T

Thatcher, Jim ; Burks, Michael R. ; Heilmann, Christian ; et al.: Web Accessibility : Web Standards and Regulatory Compliance. - 1. Auflage - New York : Springer-Verlag, 2006

W

W3C: Accessible Rich Internet Applications (WAI-ARIA) 1.0, 18.1.2011.

URL: <<http://www.w3.org/TR/wai-aria/complete>>, verfügbar am 20.4.2012

W3C: Geolocation API Specification, 10.5.2012.

URL: <<http://www.w3.org/TR/geolocation-API/>>, verfügbar am 3.8.2012

W3C: HTML5, A vocabulary and associated APIs for HTML and XHTML, 26.6.2012.

URL: <<http://dev.w3.org/html5/spec/Overview.html>>, verfügbar am 26.7.2012

W3C: Media Queries, 19.6.2012.

URL: <<http://www.w3.org/TR/css3-mediaqueries/>>, verfügbar am 22.6.2012

W3C: Mobile Web Best Practices 1.0, Basic Guidelines.

URL: <<http://www.w3.org/TR/mobile-bp/>>, verfügbar am 18.4.2012

W3C: Understanding WCAG 2.0, A guide to understanding and implementing Web Content Accessibility Guidelines 2.0.

URL: <<http://www.w3.org/TR/UNDERSTANDING-WCAG20/>>, verfügbar am 14.5.2012

W3C: Web Accessibility Initiative, WAI-ARIA FAQ.

URL: <<http://www.w3.org/WAI/aria/faq>>, verfügbar am 20.4.2012

W3C: Web Content Accessibility Guidelines (WCAG) 2.0.

URL: <<http://www.w3.org/TR/WCAG20>>, verfügbar am 14.5.2012

W3C: Web Content Accessibility Guidelines (WCAG) 2.0, Autorisierte deutsche Übersetzung.

URL: <<http://www.w3.org/Translations/WCAG20-de/>>, verfügbar am 14.5.2012

W3C: Web Style Sheets - CSS tips & tricks, Units, 4.4.2012.

URL: <<http://www.w3.org/Style/Examples/007/units>>, verfügbar am 25.6.2012

Wee, Miranda ; Lang, Tania: Writing Effective Mobile Web Content, in Peak Usability, 28.2.2011.

URL: <<http://peak-cms01.squiz.net/articles/writing-effective-mobile-web-content/>>, verfügbar am 29.2.2012

Windows Phone: UI Design and Interaction Guide for Windows Phone 7. - 2. Auflage, 2010

Wroblewski, Luke: Mobile First. - 1.Auflage - New York : A Book Apart, 2011

Wroblewski, Luke: RESS: Responsive Design + Server Side Components, 12.9.2011.

URL: <<http://www.lukew.com/ff/entry.asp?1392>>, verfügbar am 13.4.2012

Wroblewski, Luke: Why Seperate Mobile & Desktop Web Pages?, 1.9.2011.

URL: <<http://www.lukew.com/ff/entry.asp?1390>>, verfügbar am 13.4.2012

Erklärung zur selbstständigen Anfertigung

Ich erkläre, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Chemnitz, den 25. September 2012

Katharina Schönefeld